



UNICAMP

Universidade Estadual de Campinas

Faculdade de Engenharia Elétrica e de
Computação

LCSI

Departamento de Máquinas, Componentes e
Sistemas Inteligentes

**Laboratório de Controle e Sistemas In-
teligentes**

LABORATÓRIO VIRTUAL APLICADO À EDUCAÇÃO EM ENGENHARIA ELÉTRICA

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação
da Universidade Estadual de Campinas, como parte dos requisitos
exigidos para a obtenção do título de

Mestre em Engenharia Elétrica
por

Paulo Victor de Oliveira Miguel

Bacharel em Ciências da Computação - UNICAMP/SP

Prof. Dr. Gilmar Barreto

Orientador - FEEC/UNICAMP/SP

16 de Outubro de 2008

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

M588L	<p>Miguel, Paulo Victor de Oliveira Laboratório virtual aplicado à educação em engenharia elétrica / Paulo Victor de Oliveira Miguel. -- Campinas, SP: [s.n.], 2008.</p> <p>Orientador: Gilmar Barreto. Dissertação de Mestrado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.</p> <p>1. Ensino a distância. 2. Semiótica. I. Barreto, Gilmar. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.</p>
-------	--

Título em Inglês: Virtual laboratory for electrical engineering education

Palavras-chave em Inglês: Distance education, Semiotic

Área de concentração: Automação

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Ana Cristina Cavalcanti Lira , Clênio Figueiredo Salviano

Data da defesa: 16/10/2008

Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Paulo Victor de Oliveira Miguel

Data da Defesa: 16 de outubro de 2008

Título da Tese: "Laboratório Virtual Aplicado à Educação em Engenharia Elétrica"

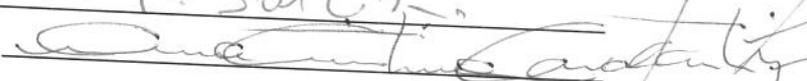
Prof. Dr. Gilmar Barreto (Presidente):



Dr. Clênio Figueiredo Salviano:



Profa. Dra. Ana Cristina Cavalcanti Lyra:





UNICAMP

Universidade Estadual de Campinas

LCSI Faculdade de Engenharia Elétrica e de
Computação
Departamento de Máquinas, Componentes e
Sistemas Inteligentes
**Laboratório de Controle e Sistemas In-
teligentes**

Tese : **LABORATÓRIO VIRTUAL APLICADO À
EDUCAÇÃO EM ENGENHARIA ELÉTRICA**

Autor : **Paulo Victor de Oliveira Miguel**

Orientador : **Prof. Dr. Gilmar Barreto**

Avaliada em 16 de Outubro de 2008 pela banca examinadora:

- Prof. Dr. Gilmar Barreto - FEEC - UNICAMP (Presidente)
- Prof. Dra. Ana Cristina Cavalcanti Lira - FEEC - UNICAMP
- Prof. Dr. Clênio Figueiredo Salviano - DMPS - CTI

Campinas, SP
2008

Resumo

Um programa para simular circuitos elétricos por computador pode ser um recurso muito útil em análise de projetos de engenharia. Para que se pudesse utilizar este recurso na formação de profissionais em engenharia elétrica, estudou-se as características específicas e criou-se então uma plataforma flexível de software. Com funcionalidades multimídia e modelos matemáticos, pode-se utilizá-la para apresentações, simulações e verificação de situações teóricas e práticas. Também pode-se utilizá-la para educação a distância inclusive via Internet, utilizando conceitos de orientação a objeto, reutilização de software e imagens vetoriais. Com o objetivo de validar a aplicabilidade do trabalho foi desenvolvida também uma metodologia para testes de usabilidade, específica para uma ferramenta de educação a distância.

Palavras-chave: Laboratório Virtual, Educação a Distância, Modelamento Matemático de Circuitos, Painel Virtual, Protoboard Virtual, Usabilidade, Semiótica, AvalUEAD.

Abstract

The computer software emulating electronic circuits can be very useful in engineering analysis. In order to use this resource in Electrical Engineering education, it has studied the necessary requirements and created a flexible software framework. Using multimedia features and mathematical modeling it can be used to make presentations, emulating processes and to show theoretical and practical situations. Additionally it could be used in electronic learning, even through the Internet, working with object oriented, reusability and vector images concepts. Even in order to test the applicability it was developed an specific usability testing for electronic learning systems.

Keywords: Virtual Laboratory, Computer Aided Education, Mathematical Scroll, Virtual Panel, Virtual Protoboard, Usability, Semiotic, AvalUEAD.

Agradecimentos

Agradeço a minha Esposa, meus Filhos e meus Pais, que tudo fizeram para que eu pudesse concluir com sucesso os trabalhos aqui desenvolvidos.

Ao meu amigo, grande profissional e orientador Prof. Dr. Gilmar Barreto e aos outros mestres e colaboradores que de alguma forma me ajudaram até a conclusão deste trabalho.

Aos colegas professores do COTUCA, que me incentivaram e apoiaram.

Aos demais colegas de pós-graduação, pelas críticas e sugestões.

Todo trabalho de pesquisa científica tem algumas características em comum. Pode-se destacar a busca pela inovação e pela perfeição através da dedicação, da persistência e do desenvolvimento do conhecimento. Um aspecto importante, no entanto, é que o resultado destes trabalhos, podem ter a participação de várias pessoas, principalmente daquelas que nos acompanham naquele momento da vida. São pessoas que direta ou indiretamente participam, ajudam, interferem e modificam nossa maneira de pensar e agir, muitas vezes permanecendo no anonimato. A estas humildemente dedico este trabalho que, com muito carinho e respeito, fizemos juntos.

Sumário

Lista de Figuras	xiii
Lista de Tabelas	xv
Lista de Algoritmos	xvii
Glossário	xix
Trabalhos do Autor Aceitos para Publicação	xix
1 INTRODUÇÃO	1
1.1 Contexto	1
1.2 Motivação	1
1.3 Objetivo	4
1.4 Planejamento	5
1.5 Conclusão deste capítulo	7
2 A EDUCAÇÃO ASSISTIDA POR COMPUTADOR	9
2.1 EAD no Brasil	9
2.2 Os Objetos de Aprendizagem e a Hipermídia	10
2.3 A Cognição e a Espiral do Conhecimento	12
2.4 Laboratórios Virtuais	14
2.4.1 O LabVirt da USP	15
2.4.2 O RIVED da SEED	16
2.4.3 Laboratório Remoto e Laboratório Virtual	16
2.4.4 Laboratório Virtual – COPPE/UFRJ	19
2.4.5 Laboratório Virtual – UNICAMP	19
2.4.6 Laboratório Virtual – Universidade de Gênova	22
2.5 Conclusão deste capítulo	22
3 MODELANDO O NÚCLEO COM A UML	25
3.1 A Busca pela Sistematização	25
3.2 O Método PDCA	26
3.3 Normas ISO/IEC	27
3.4 Desenvolvimento baseado em UML	31
3.4.1 Levantamento de Requisitos	35
3.4.2 Análise	41

3.4.3	Projeto	43
3.4.4	Implementação	45
3.4.5	Testes de Sistema	45
3.5	Conclusão deste capítulo	46
4	O DESENVOLVIMENTO DO NÚCLEO VIRTUAL	47
4.1	Diagrama Esquemático Tridimensional	47
4.2	Descrição das Camadas	49
4.3	O Núcleo do Sistema Semiótico	52
4.3.1	A Estrutura do Núcleo	52
4.3.2	Representação do Painel Virtual	56
4.3.3	O Laboratório de Corrente Alternada	57
4.3.4	Módulo de Entrada	71
4.3.5	Módulo Educador	72
4.3.6	Módulo de Monitoramento	72
4.3.7	Módulo de Saída	73
4.4	Criando um objeto sógnico	73
4.5	Estudo de Casos	74
4.5.1	Circuito Aberto	74
4.5.2	Fonte Contínua e Resistor	75
4.5.3	Fonte AC com nível DC e Resistores em Série	78
4.6	Conclusão deste Capítulo	81
5	USABILIDADE - DESENVOLVENDO O AvalUEAD	83
5.1	Método AvalUEAD	92
5.1.1	Características do Método	92
5.2	Avaliação do Sistema pelo AvalUEAD	97
5.2.1	Relatório Final de Avaliação	106
5.3	Conclusão deste capítulo	113
6	CONCLUSÕES	115
7	TRABALHOS FUTUROS	117
	Referências Bibliográficas	118
A	Métodos de Verificação de Usabilidade	127
B	Termos e Símbolos: Laboratório Virtual e Usabilidade	131

Lista de Figuras

1.1	Núcleo do Sistema	4
1.2	Apresentação da dissertação	6
2.1	Camadas Peirceanas	13
2.2	Espiral do Conhecimento	14
2.3	LabVirt-USP	15
2.4	Projeto RIVED da SEED-MEC	16
2.5	Projeto Medição Remota – UNICAMP	20
2.6	Editor de Experimentos	21
2.7	Projeto ISILab – Universidade de Gênova	21
2.8	ISILab, telas simulando equipamento real	22
3.1	UML – Diagrama Casos de Uso	37
3.2	UML – Diagrama de Classes	39
4.1	O sistema em camadas	49
4.2	Pilha Tridimensional de Implementação	51
4.3	Núcleo do Sistema	53
4.4	PV10	53
4.5	PV10 ou PV12...	54
4.6	PV12	54
4.7	PBV para PV12	55
4.8	Transformador Elétrico em um PV12	56
4.9	Sinais de entrada e saída no monitor	57
4.10	Estrutura do Laboratório AC	58
4.11	Estrutura do Painel de Conexões Virtuais	59
4.12	Estrutura do Controlador onClipEvent, tipo Round-robin	60
4.13	Repositório de Signos	64
4.14	Estrutura do Signo	65
4.15	PVÍcone	66
4.16	Resistor utilizando um PVÍcone com duas camadas, pinos 4 e 10	69
4.17	Dispositivos de Monitoração	69
4.18	Painel de Interação	70
4.19	Estrutura do Módulo de Entrada	71
4.20	Onda de ECG como sinal de Entrada	71
4.21	Módulo Educador	72

4.22	Estrutura do módulo de monitoração	73
4.23	Dispositivos de Saída	73
4.24	Circuito Aberto	75
4.25	Fonte Contínua e Resistor	76
4.26	Monitoramento de caso	77
4.27	Fonte AC com nível DC e Resistores em Série	78
4.28	Monitoramento de Caso	79
4.29	Detalhes do Painel de Monitoração	80
5.1	Gráfico AvalUEAD	112

Lista de Tabelas

5.1	<i>Checklists</i> que compõem o método <i>ErgoList</i>	86
5.2	Componentes da Técnica de Avaliação Usabilidade <i>software</i> CPqD LabiUtil	88
5.3	Componentes para Avaliação Usabilidade <i>software</i> WEB	89
5.4	Questionário de Avaliação de Usabilidade de Soluções EAD	94
5.5	Apresentação do <i>software</i>	98
5.6	Estrutura do <i>software</i>	99
5.7	Funcionalidade do <i>software</i>	100
5.8	Links e controles do <i>software</i>	101
5.9	Atratividade do <i>software</i>	101
5.10	Varredura do <i>software</i>	102
5.11	Sub-telas e Janelas do <i>software</i>	102
5.12	Interoperabilidade EAD do <i>software</i>	103
5.13	Legibilidade do <i>software</i>	103
5.14	Documentação <i>on-line</i> do <i>software</i>	104
5.15	Recursos Multimídia	104
5.16	Distância Transacional	105
5.17	Didática e Cognição	105
5.18	AVALIADOR 1 – Relatório Individual de Avaliação	107
5.19	AVALIADOR 1 – Considerações Técnicas Individuais por Componente	108
5.20	AVALIADOR 2 – Relatório Individual de Avaliação	109
5.21	AVALIADOR 2 – Considerações Técnicas Individuais por Componente	110
5.22	Laudo Final de Avaliação	111

Lista de Algoritmos

1	function calc(cvpbl,cvpbc,s1,s4,s7,s10)	61
2	function vercircuito()	62
3	function Matscroll()	63
4	<i>onClipEvent(enter frame)</i> –Processo principal do Protoboard Virtual	64
5	<i>onClipEvent(load)</i> – Instância do Signo RESISTOR	67
6	<i>fcomp()</i> – Modelamento Matemático do Signo	68
7	Passos para Criação de um Signo	74

Trabalhos Aceitos para Publicação

1. Miguel, P.V., Barreto, G. “The Core of a Semiotic Laboratory”. *AACE - Association for the Advancement of Computing in Education. E-Learn 2008, World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education*, Las Vegas - Nevada - USA, Novembro 2008.
2. Miguel, P.V., Barreto, G. “Modelo Semiótico Virtual Hierárquico”. *CREAD2008 - Congresso Internacional de Educação a Distância. Escola Nacional de Saúde Pública Sergio Arouca - ENSP/Fiocruz, Nova Southeastern University, Fischler Graduate School of Education and Social Services*, Rio de Janeiro - Brasil, Outubro 2008.
3. Miguel, P.V., Barreto, G. “Um Laboratório Virtual de Signos Semióticos”. *COBENGE - Congresso Brasileiro de Educação em Engenharia. Associação Brasileira de Educação em Engenharia - ABENGE*, São Paulo - Brasil, Setembro 2008.
4. Miguel, P.V., Barreto, G. “Laboratório Virtual de Eletrotécnica”. *SBPC 2008 - Sociedade Brasileira para o Progresso da Ciência*, São Paulo - Brasil, Julho 2008.

Capítulo 1

INTRODUÇÃO

1.1 Contexto

Pode-se utilizar a imaginação e a criatividade para mostrar, com muita riqueza de detalhes, a ficção. Com o auxílio do conhecimento e os recursos tecnológicos pode-se também mostrar e ensinar a realidade. Ferramentas virtuais adequadas podem ainda oferecer, aos estudantes e pesquisadores, ambientes adequados para exploração e testes de situações hipotéticas, até mesmo longe dos grandes laboratórios. Tornar o ensino acessível significa permitir uma melhor capacitação profissional, principalmente para os que não estão próximos dos grandes centros de formação. Como recurso adicional, as ferramentas de estudo assistido por computador, podem aumentar as interações dos alunos com as fontes de informação, os educadores e seus colegas. O relacionamento pode facilitar a inclusão do estudante e motivar o aprendizado nas mais diversas áreas do estudo e da pesquisa. Nesse contexto o desafio é fazer uso de outras ferramentas, além das tradicionais, para levar a informação e o conhecimento ao cotidiano dos nossos estudantes, facilitando a assimilação dos conceitos teóricos e práticos, promovendo ainda mais o interesse pelo estudo da Engenharia Elétrica.

1.2 Motivação

A utilização de modelos teóricos na análise de projetos de engenharia, normalmente, reduz o tempo e o custo para verificação de requisitos e validação dos resulta-

dos esperados. A implementação adicional de modelos experimentais pode enriquecer estes estudos, já que permitem considerar outros parâmetros e variáveis difíceis de serem testados somente com uma abordagem teórica.

A questão frequente é que necessitam de tempo e investimentos adicionais com planejamento, aquisição de componentes, partes e ambientes que podem até ser destruídos após cada hipótese verificada.

Utilizar a Semiótica como base teórica para o desenvolvimento de novas ferramentas destinadas ao ensino, parece ser uma idéia muito atraente [Not95]. A modelagem computacional, utilizando recursos de multimídia e programas simuladores, pode ser considerada um aprimoramento da análise puramente teórica, com uma boa relação entre benefícios e investimentos necessários [Dal93, Cob76]. Em um ambiente virtual, estes sistemas ainda tem as vantagens de reutilização e flexibilidade de configuração. Estas características tornam esta abordagem ainda mais interessante para aplicações nas áreas educacionais, devido à utilização de um ambiente dinâmico e a disponibilidade de ferramentas com muitos recursos ilustrativos e elucidativos. Se desenvolvida utilizando conceitos e tecnologias perfeitamente adaptadas ao uso em redes de computadores, principalmente através da Internet ou outros dispositivos portáteis, esta modelagem computacional pode ainda ser utilizada para educação a distância ampliando assim a disponibilidade e o alcance geográfico da informação e do conhecimento.

Este trabalho apresenta uma plataforma interativa que, utilizando bibliotecas de componentes, permite que estes sejam arranjados espacialmente e associados uns aos outros.

Representado em diagramas de blocos e esquemas de circuitos, se vinculados a modelos matemáticos, estes sistemas virtuais buscam reproduzir as características elétricas e suas variações.

Para utilização em cursos de Engenharia Elétrica, estes sistemas permitem ainda a observação de efeitos emulados de outros parâmetros dos componentes eletroeletrônicos e seus efeitos eletro-magnéticos, por exemplo.

De maneira também análoga, possibilita ainda o processo reverso de busca da causa a partir do efeito, apresentando desvios de sinais ou mesmo a ausência deles, para que os estudantes possam tentar identificar os conceitos teóricos envolvidos nas simulações e associá-los a situações reais.

Não obstante, esta solução pode ser utilizada ainda em conexão com ambientes não virtuais captando sinais, processando e acionando outros sistemas.

Como forma de validação, as interfaces utilizadas em laboratórios virtuais ou remotos, precisam ser submetidas a testes específicos de Usabilidade que considerem

aspectos pedagógicos importantes para interfaces destinadas ao apoio a Educação. Nesta análise as principais dificuldades podem vir de sistemas que implementam interfaces pouco amigáveis ou lentas. Com frequência observa-se a busca pela qualidade a um custo muito alto, utilizando-se soluções de software muito complexas e imagens de alta definição muito pesadas, com grande esforço computacional para serem processadas. O problema pode se tornar ainda pior ao se utilizar redes de comunicação de dados incapazes de suportar um alto nível de interação entre todos os envolvidos. Como resultado verifica-se, normalmente, sistemas densos com grande variedade de opções porém com apresentação didática limitada, com pouca flexibilidade de expansão, de difícil portabilidade e com pouca liberdade para as criações dos usuários.

Como desafios podemos indagar:

- Como implementar uma plataforma virtual para conectar componentes tão diferentes, como transistores, transformadores, geradores e motores elétricos dentre outros?
- Como modelos matemáticos podem ser utilizados para emular dispositivos não lineares e outros lineares no mesmo circuito?
- Como podemos mostrar alguns conceitos como indução e efeitos de ondas eletromagnéticas em alguns dispositivos de saída virtuais?
- Como fazer tudo isso sem limitar drasticamente a capacidade de criar.

De fato, todas estas questões precisam estar associadas aos requisitos:

- Interfaces amigáveis,
- Agilidade nas interações, “*fast interaction*”,
- “*Softwares*” e imagens leves,
- Incluindo ainda uma excelente qualidade de apresentação e conceitos didáticos.

Tudo isso para exercitar o raciocínio e a imaginação dos estudantes de graduação.

Em busca destas respostas este trabalho propõe um modelo denominado tridimensional para uma abordagem sistemática e estruturada de soluções destinadas ao apoio a educação. Especificamente detalha-se o projeto do Núcleo do Sistema Virtual, como uma das camadas do modelo tridimensional proposto. A arquitetura do núcleo, figura 1.1 é composta de dispositivos de entrada e saída conectados em um laboratório

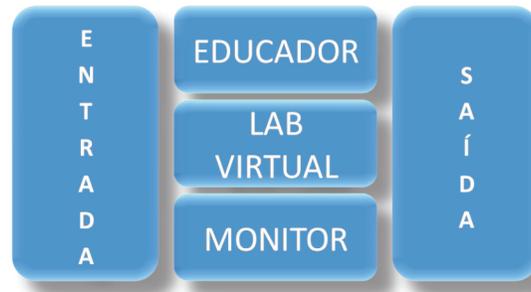


Fig. 1.1: Núcleo do Sistema

virtual. Incluiu-se também o sistema educador que propõe condições de teste e o ambiente educacional, bem como as medições a serem observadas, utilizando-se para isso do Sistema de Monitoração onde encontram-se equipamentos como multímetros e analisadores virtuais.

Todos estes módulos são encontrados no Núcleo do Sistema que é parte de uma plataforma de um modelo tridimensional. A plataforma em três dimensões é assim caracterizada por dispor no sentido vertical de camadas que sugerem a formação do conhecimento segundo a teoria semiótica de Peirce [Not95]. No sentido horizontal o modelo sugere níveis compostos de módulos e recursos específicos. No caso do núcleo verifica-se o desenvolvimento de ferramentas, equipamentos e signos de forma a tratar as áreas do conhecimento a serem estudadas ou testadas.

Todos os módulos virtuais para desenvolver e testar, de acordo com as definições teóricas, estão presentes também no Núcleo do Sistema.

O LVCCA (Laboratório Virtual para Circuitos de Corrente Alternada) considera todas as dificuldades apresentadas e propõe uma solução versátil com interfaces amigáveis para serem utilizadas em cursos de Engenharia Elétrica, Mecânica, Química, Alimentos, Agrícola e de Computação.

1.3 Objetivo

A perspectiva de se poder simular situações práticas, utilizando-se componentes virtuais em ambientes controlados e ainda com muitos recursos, só é menos atraente do que a possibilidade de disponibilizar uma base de conhecimento associada a simples e versáteis ferramentas educacionais, ampliando o alcance, facilitando o acesso à educação e ao ensino de qualidade, necessitando para isso apenas de uma solução de baixo custo.

A busca pela “*Distância Transacional*” ideal [Cas05], conceito muito utilizado nos estudos de *Educação a Distância*, está relacionada com a quantidade e a qualidade das interações entre educadores, estudantes e os recursos educacionais.

A despeito da inovação, dos desafios tecnológicos e da busca pela melhoria contínua, nada pode ser mais gratificante do que a melhoria na capacidade de ensinar, da oportunidade da inclusão social e do acesso ao conhecimento que é de fato o objetivo principal deste trabalho, motivando e auxiliando assim estudantes e educadores nos cursos de Engenharia dentre outros.

1.4 Planejamento

1. Nesta Tese apresenta-se uma ferramenta a ser aplicada ao ensino de Engenharia Elétrica. Verifica-se portanto a abrangência do projeto e identifica-se a implementação necessária para a validação de uma ferramenta a ser aplicada ao ensino. A apresentação de conceitos relacionados à Educação Assistida por Computador e com Laboratórios Virtuais será realizada no Capítulo 2 desta Tese.
2. Esta ferramenta foi projetada, implementada e validada em exemplos tradicionais de Eletrotécnica. Na figura 1.2 apresenta-se um organograma estrutural desta tese. Utilizando um modelo de desenvolvimento de “*software*” buscase identificar os requisitos, estruturar as informações e organizar os recursos necessários.
3. Desta forma pretende-se apresentar uma ferramenta com resultados práticos, as avaliações efetuadas e as propostas para novos trabalhos. Os capítulos desta Tese estão relacionados a seguir.

Os capítulos estão organizados da seguinte forma:

- Capítulo 1: descreve o contexto, a motivação e os objetivos envolvidos neste projeto.

Com o desafio e a responsabilidade de apresentar novas alternativas para auxiliar a educação e de facilitar o desenvolvimento de conteúdo didático-eletrônico, descreve a idéia, as etapas e as metas almejadas com esta Tese.

- Capítulo 2: apresenta alguns conceitos relacionados à Educação Assistida por Computador e a Laboratórios Virtuais.



Fig. 1.2: Apresentação da dissertação

Utilizando estudos desenvolvidos em renomados centros de estudo e pesquisa, mostra exemplos de trabalhos relacionados com o tema “Laboratórios Virtuais”. Utilizando-se ainda de uma literatura recente relacionada à Educação a Distância, apresenta o conhecimento utilizado na elaboração da Tese.

- Capítulo 3: descreve o planejamento e o modelamento do projeto.

Utilizando um modelo reconhecido para desenvolvimento de projetos e o conhecimento em melhoria de qualidade contínua, projeta a ferramenta que será desenvolvida e submetida aos testes.

- Capítulo 4: apresenta o detalhamento e o desenvolvimento do sistema.

A partir do diagrama funcional de cada componente do Núcleo do Sistema descreve os recursos e suas funcionalidades. Identifica ainda resultados e demonstra sua potencialidade nas possíveis áreas de aplicação.

- Capítulo 5: destacando-se a importância do tema USABILIDADE associado a uma ferramenta educacional, incorpora e propõe uma ferramenta para validar uma solução de software dedicada à Educação a Distância. Efetua portanto os testes, com o objetivo de atender as recomendações de ergonomia e principalmente de USABILIDADE.
- Capítulo 6: descreve as conclusões relativas à implementação efetuada.
- Capítulo 7: identifica as alternativas de novos trabalhos e as apresenta de forma objetiva.

- Referências Bibliográficas
- Apêndices:
 - Métodos de Verificação de Usabilidade.
 - Termos e Símbolos: Laboratório Virtual e Usabilidade.

1.5 Conclusão deste capítulo

Neste capítulo foi feita a apresentação geral da tese. A partir do contexto em que foi elaborada destacou-se a motivação em se desenvolver ferramentas de inclusão cultural e social, dispositivos de apoio presencial e uma estrutura padronizada que facilite o desenvolvimento de “Objetos de Aprendizagem” em diversas áreas da ciência e da educação. Com o objetivo de se testar a usabilidade de uma solução implementada utilizando-se uma metodologia reconhecida, efetuou-se o planejamento desde os conceitos teóricos até os testes que dão suporte às conclusões finais. No capítulo seguinte, dá-se início aos fundamentos teóricos para o desenvolvimento dos trabalhos, a associação com a Semiótica e seus conceitos de aprendizagem. Destacam-se ainda outras iniciativas semelhantes e as oportunidades de contribuição a serem desenvolvidas neste trabalho.

Capítulo 2

A EDUCAÇÃO ASSISTIDA POR COMPUTADOR

2.1 EAD no Brasil

A Educação a Distância é um grande movimento em forte expansão no Brasil. Muitas instituições tem projetos próprios ou em conjunto com outras instituições, com o objetivo de desenvolver ferramentas e ambientes que possam ampliar o alcance dos cursos oferecidos ou ainda para servir como ferramenta de apoio ao cursos presenciais. Por outro lado o Ministério da Educação (MEC), por meio da Secretaria de Educação a Distância (SEED) [MEC08] atua como um agente de inovação tecnológica nos processos de ensino e aprendizagem, fomentando a incorporação das Tecnologias de Informação e Comunicação (TICs) e das técnicas de educação a distância aos métodos didático-pedagógicos. Além disso, promove a pesquisa e o desenvolvimento voltados para a introdução de novos conceitos e práticas nas escolas públicas brasileiras. Buscando uma uniformidade de interpretação e a regulamentação, a Educação a Distância foi definida como a modalidade educacional na qual a mediação didático-pedagógica nos processos de ensino e aprendizagem ocorre com a utilização de meios e tecnologias de informação e comunicação, com estudantes e professores desenvolvendo atividades educativas em lugares ou tempos diversos. Essa definição está presente no Decreto 5.622, de 19.12.2005 (que revoga o Decreto 2.494/98), que

regulamenta o Art. 80 da Lei 9394/96 (LDB) [SEE08].

É importante salientar no entanto que além dos esforços de inclusão digital, fundamentais para permitir a utilização destas novas tendências por todos os cidadãos, é necessário que se intensifique o investimento em comunicação visando disponibilizar o acesso em alta velocidade a todos os potenciais usuários destes novos sistemas em todas as regiões do país.

2.2 Os Objetos de Aprendizagem e a Hipermissão

Uma definição simples porém de fácil entendimento é a de que um Objeto de Aprendizagem é um recurso digital (imagem, filme, etc.) que pretende ser utilizado para fins pedagógicos e que possui, internamente ou através de associação, sugestões sobre o contexto apropriado para sua utilização [Sos02].

Os “Objetos de Aprendizagem” podem trazer em seu contexto, direta ou indiretamente, áreas diversas como a filosofia, a semiótica, a psicologia, a antropologia, a computação gráfica, a animação, as engenharias, as ciências cognitivas, a publicidade, o marketing, as comunicações, o design, as críticas literária e artística além da narratologia dentre outras. Classificados como Híbridos [Men04], estes, os Objetos de Aprendizagem envolvem programação, roteiro de navegação, design de interface, técnicas de animação e usabilidade. Manifesta-se portanto a natureza intersemiótica da Hipermissão [Men04], a constelação e a intersecção de linguagens ou processos sígnicos que neles se concentram.

Pode-se destacar a interação e a imersão como causa e efeito de ferramentas aplicadas à educação [Cas05]. A interatividade é uma propriedade intrínseca da comunicação digital [San04], e está relacionada com ações reativas do aluno que se dão dentro de parâmetros que são as regras e instruções estabelecidas pelas variáveis da ferramenta educacional. Mas pode haver também uma alta interatividade em soluções de maior complexidade, multiplicidade, não-linearidade [Men04], bi-direcionalidade, potencialidade, permutabilidade (combinatória), imprevisibilidade etc., permitindo ao aluno uma maior participação e criatividade. A interatividade não é caracterizada apenas como experiência ou agenciamento do estudante, mas como possibilidade de co-criação de uma obra aberta e dinâmica, em que os experimentos e novas situações de aprendizagem se reconstróem a cada lição, a medida que se utilizam signos na formação do conhecimento.

A imersão é outra propriedade esperada da comunicação digital, ela deve se acentuar com identificação de um espaço simulado e a possibilidade do aluno ser envolvido por

esse espaço como na realidade virtual, principalmente com a utilização dos objetos de aprendizagem, onde se pode alcançar altos níveis de imersão, constituindo-se assim no paradigma da construção espacial do laboratório virtual – a geração de objetos e ambientes tridimensionais configuráveis através da modelagem de componentes e dispositivos. Além dessa aparência visual, a simulação visa modelar de maneira pedagógica como os objetos, substâncias, componentes e equipamentos agem, reagem, movem-se, aquecem-se, e modificam suas características estruturais, químicas, físicas e eletro-mecânicas que desejamos estudar.

Pode-se definir a hipermídia como uma linguagem complexa e híbrida, que possui cinco características fundamentais:

1. Hipertextualidade é a forma fragmentada e descentralizada de distribuição das informações num dado ambiente virtual, caracterizada pelo hipertexto, possibilita a leitura não seqüencial do conteúdo e a flexibilidade no acesso às informações pertinentes.
2. Navegabilidade é a capacidade de se mover no espaço virtual. Além de disponibilizar vários caminhos, deve apresentar variações na forma de navegação.
3. A não-linearidade, está relacionada com a mobilidade do usuário que não impede a escolha dos caminhos a serem percorridos, apesar de um fluxo de navegação bem sinalizado. Fornece a impressão de que há sempre mais informações a serem descobertas pelo usuário.
4. Hibridismo é a característica de uma interface que permite integrar imagens, ilustrações, fotografias, animação e som originando um novo contexto visual e semântico.
5. Interatividade é a capacidade que o aluno tem de acessar, estabelecer relações e interferir nos documentos, modificando as informações, configurando, e participando na formação do conhecimento. Ao contrário de uma simples animação, em um sistema interativo, o aluno pode seguir sentidos distintos para o mesmo programa, não sendo possível para o autor do sistema prever a multiplicidade de sentidos possíveis seguidos durante a utilização da ferramenta de aprendizagem.

Apesar de já habituado com esse universo eletrônico, o uso de hipermídias na criação de ambientes com atividades pedagógicas, coloca o aluno frente a situações de aprendizado co-participativas ampliando desta forma seu senso crítico.

O aluno atual, membro da revolução tecnológica que presenciamos atualmente, possui uma maior proximidade com a linguagem visual do que com a escrita [Men04], facilitando assim, a introdução de novos recursos educacionais como por exemplo, os assistidos por computador. Como ferramenta adicional, os laboratórios virtuais podem oferecer alternativas para que os educadores, utilizando outras linguagens, possam despertar nesse estudante moderno um interesse ainda maior pelos estudos. A Hipermídia pode cumprir importante papel se transformada em aliada do processo educacional, se usada com critério e se proporcionar a reflexão e o desenvolvimento da criticidade do aluno.

2.3 A Cognição e a Espiral do Conhecimento

O termo Terceiridade, não se refere à experiência ou ao amadurecimento do processo de Educação Assistida por Computador, mas sim ao modelo proposto por Charles Sanders Peirce, aqui apresentado de forma muito resumida e objetivamente descrito para situar os trabalhos desenvolvidos nesta tese.

Peirce dizia que a única coisa que se encontra em expansão é a mente humana, sendo assim tudo já está no mundo, e portanto se algum fenômeno não está explícito é porque nossa mente ainda não consegue enxergá-lo ou explicá-lo. É necessário então expandir a compreensão humana para tudo o que já foi escrito sobre linguagem. A tese central de Peirce é a de que todo pensamento se dá em signos, do que decorre que a cognição é triádica, figura 2.1, formada por três termos básicos, uma relação entre um sujeito e um objeto sempre mediada pelo signo [Not96, Not95]. A Primeiridade (firstness), está relacionada com as idéias vagas, com o sentimento, ou seja, a apreensão inicial das coisas, são partes integrantes da consciência, da sensação e do pensamento. Pode-se assim definir um quali-signo [Not95], por exemplo, como uma primeira forma imprecisa e indeterminada de predicação das coisas, relacionados com o sentimento, o acaso, o indeterminado, as coisas vagas, um conjunto de informações ainda não associadas, uma novidade. Uma mônada, que não tem relação com nada totalmente determinado, é a pura possibilidade, o começo de um processo de semiose que poderá continuar ou não. Para efeito desta tese, consideraremos inicialmente uma base de informações e idéias que podem suscitar a Primeiridade em um processo de aprendizagem.

A Secundidade (Secondness), se apresenta como uma continuação da Primeiridade no processo de cognição proposto por Peirce [Not96], uma sensação explicada pela reação, pela existência, pelo dependente, relativo, o aqui-agora, o determinado, a

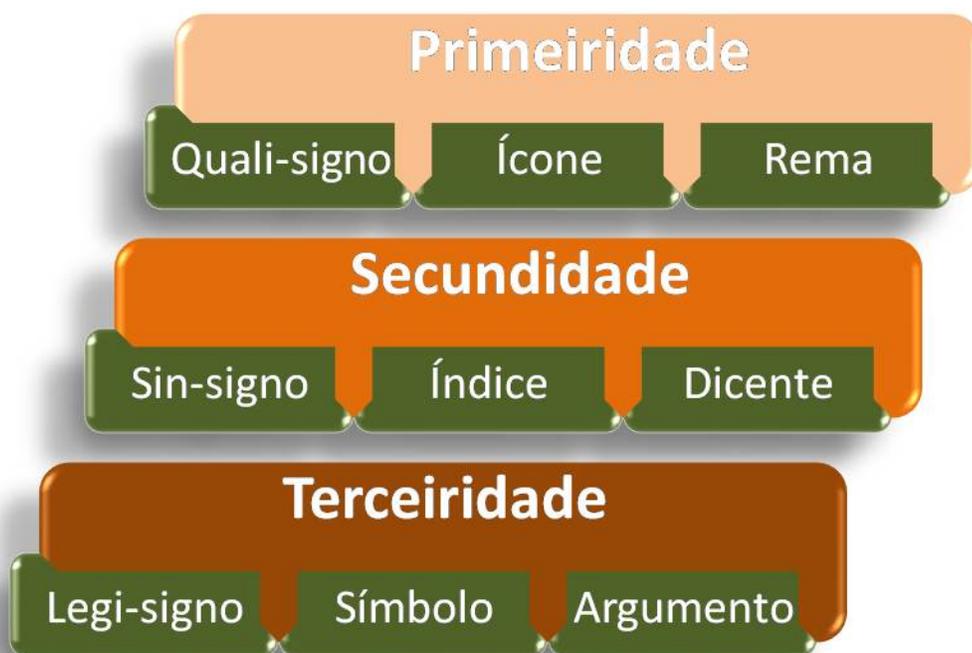


Fig. 2.1: Camadas Peirceanas

polaridade, a ação-reação, um mundo real, sensual e pensável. Nesse contexto, pretendemos suscitar a Secundidade através da associação das informações, do planejamento, dos diagramas e principalmente da interconexão das idéias. Certamente, onde quer que haja um fenômeno, há uma qualidade, e isto é, sua Primeiridade. Não obstante, a qualidade é apenas uma parte do fenômeno, visto que, para existir, a qualidade tem que estar presente em algum componente de aprendizagem.

A proposta Triádica de Peirce nos leva ainda à Terceiridade (Thirdness), que significa signo, continuidade, semiose, aprendizagem, cognição, tempo, mediação, lei, mente e se encontra no terreno da razão e da tríade. Complementa a secundidade que, por sua vez, necessita das ações diádica, embutida dentro da ação [triádica] do signo, e da ação inteligente ou semiose, da mesma forma que associamos os signos às coisas que são também signos, e dos signos que só podem ser signos porque são também coisas. A razão, como parte superficial da consciência é, de acordo com as categorias propostas por Peirce, um terceiro momento da apreensão e compreensão de um fenômeno. A Terceiridade, a mais inteligível para nós, é a síntese intelectual ou o pensamento em signos, a medição entre nós e o mundo, é o terreno do pensamento. Identificamos aqui grandes oportunidades para os objetos de aprendizagem que, ricos em signos e se implementados segundo os conceitos apresentados, podem ser classificados como ferramentas facilitadoras triádicas utilizadas no processo de cognição.

Sendo a mente humana o verdadeiro universo em expansão para o conhecimento, ferramentas virtuais projetadas à luz da teoria do conhecimento, que acumula portanto esforços de estudiosos como Aristóteles, Platão, Locke, Kant, Berkeley, Suassure, Hjelmsleve, Peirce e outros, podem intensificar a espiral semiótica de formação do conhecimento onde um objeto está relacionado com um signo e este com seu interpretante e ainda este, por conseguinte, com o objeto que já não é mais o mesmo, modificado pelo processo de cognição, realimentando uma espiral virtuosa de formação do conhecimento, figura 2.2.



Fig. 2.2: Espiral do Conhecimento

2.4 Laboratórios Virtuais

Utilizando-se recursos tecnológicos e a criatividade podemos manipular a realidade virtual de forma a cumprir um papel pedagógico caracterizado pelos objetos de aprendizagem. Uma modalidade destes objetos são os Laboratórios Virtuais, estes surgiram com a necessidade de diversificar a criação de material didático, em especial, com o uso das TICs (Tecnologia da Informação e Comunicação) e software especiais, proporcionando o acesso a experiências que facilitem a consolidação dos conceitos. Na busca para a construção de modelos e conseqüente aquisição de experiência com objetos de aprendizagem, ocorreram iniciativas e investimentos por parte das instituições de ensino públicas e privadas. Cabe ressaltar aqui a LDB (lei de diretrizes e

bases da educação) que em seu artigo 80 cita a inserção da EAD no contexto educacional Brasileiro e que motivou diversificadas pesquisas e implementações.

2.4.1 O LabVirt da USP

O Laboratório Didático Virtual (LabVirt) [IU08] é uma iniciativa da Universidade de São Paulo (USP), atualmente coordenado pela Faculdade de Educação. Trata-se de um projeto que busca despertar no aluno o pensamento crítico, o uso do método científico, o gosto pela ciência e principalmente a reflexão e compreensão do mundo que o cerca. Partindo de simulações feitas pela equipe do LabVirt pode-se utilizar “projetos educacionais” interessantes na Internet bem como respostas de especialistas para questões enviadas através do site. O LabVirt tem como principal objetivo construir uma infra-estrutura pedagógica e tecnológica que facilite o desenvolvimento de projetos de física e química na escolas, figura 2.3.

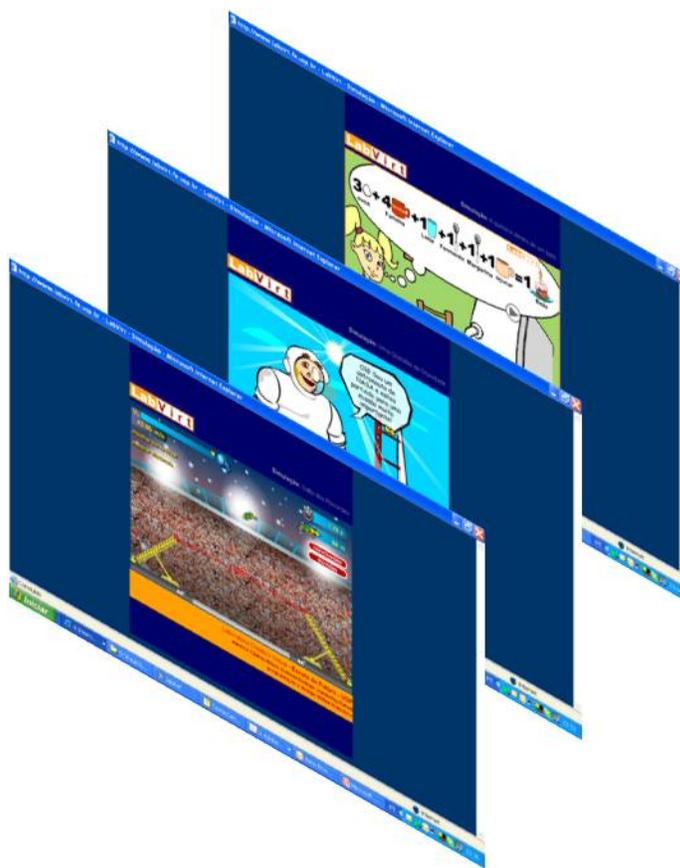


Fig. 2.3: LabVirt-USP

2.4.2 O RIVED da SEED

O RIVED [RIV05] é uma iniciativa da Secretaria de Educação a Distância (SEED). A exemplo do LabVirt da USP, descrito anteriormente, tem por objetivo a produção de conteúdos pedagógicos digitais, na forma de objetos de aprendizagem e busca estimular o raciocínio e o pensamento crítico dos estudantes, associando o potencial da informática às novas abordagens pedagógicas, melhorar a aprendizagem das disciplinas da educação básica e a formação da cidadania no indivíduo. O RIVED, figura 2.4, promove também a capacitação sobre a metodologia para produzir e utilizar os objetos de aprendizagem nas instituições de ensino superior e na rede pública de ensino. Além disso promove a produção e publicação na WEB dos conteúdos digitais para acesso gratuito.

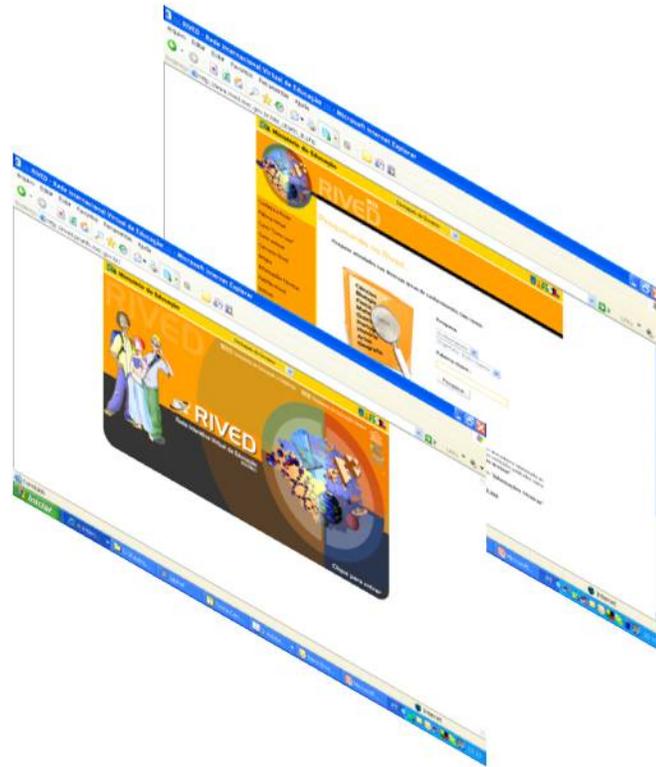


Fig. 2.4: Projeto RIVED da SEED-MEC

2.4.3 Laboratório Remoto e Laboratório Virtual

Projetos de Laboratórios Remotos são feitos [Mar07] com o objetivo de desenvolver uma plataforma de suporte a experiências que ocorrem em vários ambientes distintos, implementados a partir de laboratórios reais atendendo a características

como:

- Redução de custo operacional por aluno com o compartilhamento de recursos,
- Modularidade,
- Execução dos experimentos em “tempo real”,
- Utilização dos procedimentos experimentais, facilitando a integração com livros-texto e outros,
- Compartilhamento de ambientes e equipamentos de alto custo,
- Flexibilização de horários para execução dos experimentos.

Cabe relevar que Laboratório Remoto, neste caso, refere-se a laboratórios reais automatizados que podem ser acessados no próprio local ou via Internet. Vamos então prosseguir, mostrando a importante diferença entre os conceitos de Laboratório Remoto e Laboratório Virtual. Esta diferença é um ponto determinante para esta tese já que, apesar de serem estruturas complementares, a semiologia [Not95] nos permite desafiar e indagar: Quais são os verdadeiros limites da cognição, da formação do conhecimento dentro do mundo Virtual? De fato tudo passa pela definição/utilização dos signos, da dúvida sobre a completa apreensão dos objetos pela mente humana, algumas das questões que vêm incomodando o estudo do conhecimento desde Platão e Aristóteles.

Os Laboratórios acessíveis via rede de comunicação, local ou remota, desenvolveram-se a partir da generalização do uso das novas tecnologias da informação e comunicação (TICs). A consolidação do uso da Internet como ferramenta de compartilhamento de informações, e o desenvolvimento de ferramentas modernas de controle, aquisição e distribuição de dados via rede de computadores vêm impulsionando o controle e o acompanhamento de ensaios experimentais remotos [Mar07].

Um Laboratório Remoto existe fisicamente em algum lugar como um laboratório de eletrônica real, enquanto que o Laboratório Virtual pode ser um sistema de computação que se utiliza de Objetos de Aprendizagem [Sos02], para o desenvolvimento de experiências, testes e validação de proposições.

Um Laboratório Remoto é formado por um conjunto de instrumentos, medidores e/ou atuadores com ou sem processamento local, mas que são controlados segundo instruções determinadas remotamente a partir de outros ambientes. O usuário remoto acessa e controla o laboratório e, a partir daí, aciona equipamentos, faz observações, testa condições e coleta dados. Além disso pode-se capturar imagens do ambiente

remoto, com o uso de câmeras de vídeo, ampliando os ensaios e distribuindo informações em tempo real.

Em um laboratório de engenharia elétrica pode-se monitorar fenômenos como:

- características eletromagnéticas, corrente e tensão,
- frequência e formas de onda de sinais específicos,
- temperatura, pressão, volume, deslocamento,
- erros e defeitos resultantes de alterações intencionais ou não, dentre outros.

Um Laboratório Virtual é proposto, normalmente, com os mesmos objetivos de um laboratório real, porém, em geral, utilizando recursos diferentes. São formados por configuradores, simuladores e monitores virtuais que podem ou não evoluir para um ambiente real. No entanto, podem existir ainda algumas vantagens como ensaios destrutivos e verificação de situações críticas onde se busca identificar os limites de determinadas características, sem que com isso haja o perigo de se danificar um instrumento ou de se atentar contra a sua própria segurança. Pode-se ainda simular atividades em ambientes ainda não implementados fisicamente, efetuar o treinamento para atividades de alto risco, como a realização de experiências com altas tensões, por exemplo.

Pode-se destacar aqui alguns componentes operacionais inerentes a soluções onde exista um ambiente físico associado:

- Controle de acesso dos usuários ao ambiente físico a partir do ambiente remoto,
- Software de agendamento para alocação de recursos físicos,
- Hardware de interface para controle e/ou supervisão no ambiente remoto,
- Software e sistema de comunicação para interação com ambiente remoto.

Não são componentes simples, principalmente se desejarmos um laboratório versátil e uma ampla rede de usuários, o que aliás são alguns dos objetivos usuais deste tipo de projeto. Estas são ainda características que podem influenciar em tempo de desenvolvimento, custo, flexibilidade e versatilidade destas soluções.

2.4.4 Laboratório Virtual – COPPE/UFRJ

Projeto da UFRJ que visa controlar a alocação e o compartilhamento de bancadas virtuais que integram os projetos PAPROS e PALAS, componentes da Teia de Interconexão de Laboratórios Virtuais [Sil05]. O Projeto TEIA visa criar um ambiente didático através da participação dos alunos de forma cooperativa com os professores que darão apoio didático aos laboratórios, visando motivar e acelerar o aprendizado, solidificando os conhecimentos adquiridos pelos alunos através da elaboração dos conteúdos complementares e abertos dos módulos componentes dos laboratórios virtuais. A construção da teia envolve os laboratórios:

- PACUS: Páginas de apoio para o ensino de Cálculo, incluindo os aspectos de limite, diferenciação, integração, seqüências e séries;
- PALAS: Páginas de Apoio a um Laboratório de Sistemas Lineares, visa dar um apoio interativo para o ensino da teoria fundamental de sistemas lineares;
- PALIN: Páginas de apoio para o ensino de Álgebra Linear, incluindo os aspectos vetoriais de dimensão finita e os espaços de funções;
- PATROL: Páginas de apoio para o ensino de Sistemas de Controle, incluindo aspectos de controle clássico e moderno;
- PAPROS: Páginas de apoio para o ensino de Processamento de Sinais, incluindo aspectos do processamento de sinais e sistemas discretos e analógicos;
- PANN: Páginas de apoio para o ensino de Redes Neurais, incluindo aspectos dos treinamentos supervisionado e não supervisionado e aplicações em física experimental e engenharia.
- Módulo de ferramentas: Páginas de apoio ao desenvolvimento técnico de sistemas de software para Internet.

Observa-se um grande esforço em propiciar uma autenticação eficiente dos usuários e um sistema de alocação dos recursos já que a interface com o laboratório real é feita com o LabView.

2.4.5 Laboratório Virtual – UNICAMP

Um sistema [Che04] desenvolvido com o propósito de efetuar captação de dados efetuando medições remotas e com aplicação educacional, utilizando-se para isso

de instrumentos de medição, placas para instrumentação e os cabos e interfaces para conexão. Para que seja possível estabelecer comunicação com o hardware dos instrumentos e placas foram utilizados padrões e protocolos de comunicação como o GPIB. Ambientes de programação como o Labview e C++, permitiram acessar e controlar o fluxo de dados desses equipamentos, possibilitando o armazenamento e visualização de informações. Para o desenvolvimento de experimentos na área de engenharia elétrica/eletrônica foi implementado neste trabalho um sistema de hardware constituído por uma placa eletrônica de controle, um barramento de dados e as placas de experimentos:

- Comunicação com o computador por meio da porta USB2.
- Cada placa de experimento pode controlar até 32 relés, 8 potenciômetros digitais e comutar até 8 canais de instrumentos de medição.
- Controle de até 8 experimentos simultâneos.

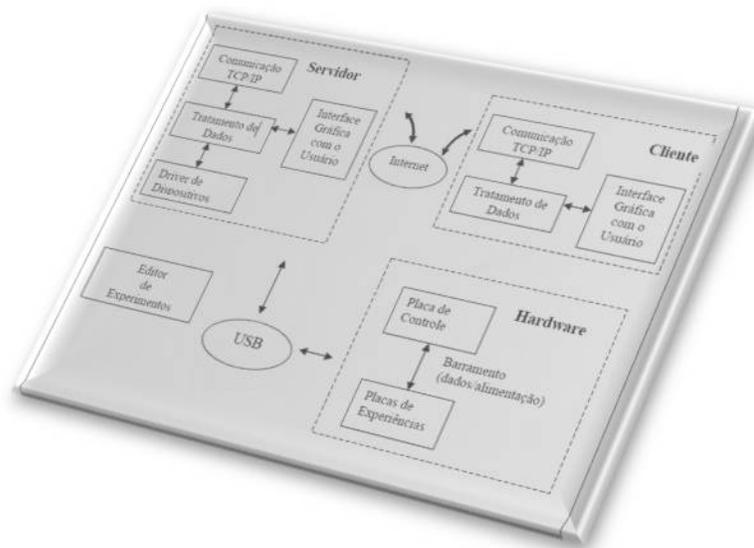


Fig. 2.5: Projeto Medição Remota – UNICAMP

Além do uso do LabView pode-se mencionar a estrutura Cliente/Servidor figura 2.5, mais uma vez utilizada como organização das camadas remotas, de rede e do ambiente de medição. Pode-se também destacar um editor de experimentos figura 2.6.

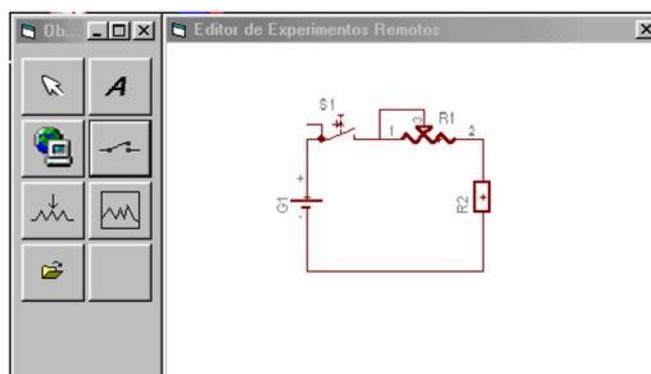


Fig. 2.6: Editor de Experimentos

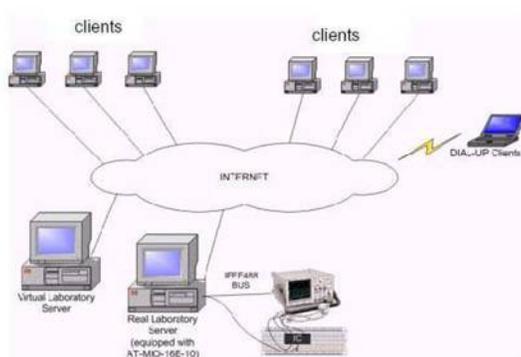


Fig. 2.7: Projeto ISILab – Universidade de Gênova

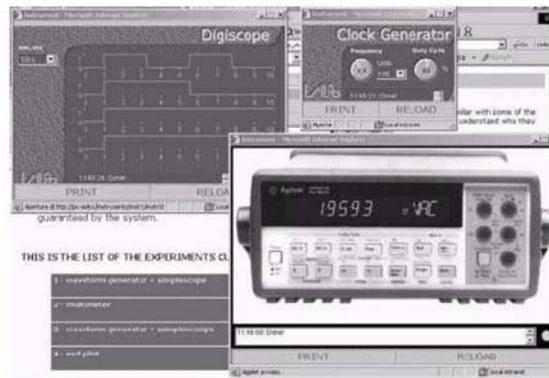


Fig. 2.8: ISILab, telas simulando equipamento real

2.4.6 Laboratório Virtual – Universidade de Gênova

O ISILab [Bag02] é um projeto desenvolvido pelo DIBE (Department of Biophysical and Electronic Engineering) da Universidade de Gênova, Itália. Pode-se destacar inicialmente uma implementação em camadas, os Servidores Virtual (VLS) e o Real (RLS), figura 2.7, a arquitetura cliente/servidor e o uso do LabView como principal interface no RLS. Também a tentativa de simulação através do uso da Realidade Virtual, figura 2.8, onde as telas virtuais simulam um modelo específico de analisador de estados lógicos (DIGISCOPE) e um gerador de pulsos (AGILENT).

2.5 Conclusão deste capítulo

Neste capítulo fez-se considerações relativas à aplicação do computador na educação e algumas iniciativas nacionais e internacionais para o desenvolvimento de ferramentas com este objetivo. Após a descrição dos principais órgãos oficiais associados à Educação a Distância no Brasil, destacou-se a importância para o desenvolvimento e a inclusão cultural no país. Ao se apresentar os “Objetos de Aprendizagem” e suas peculiaridades introduziu-se a Semiótica de Peirce baseada nos “Signos” triádicos e a espiral de formação do conhecimento. Prosseguindo, observou-se ainda a limitada oferta de programas e produções de qualidade com bom conteúdo pedagógico, apesar do grande esforço em se desenvolver novas soluções e novos temas. O grande destaque ficou com as iniciativas para se organizar equipes de profissionais de informática, educadores e entusiastas de forma a promover a formação e o desenvolvimento de novos aplicativos educacionais. Nas soluções de Laboratórios Virtuais apresentadas verificou-se o foco em alguma característica específica, soluções destinadas a experimentos e demonstração de proposições teóricas contam com poucos modelos de

desenvolvimento e de interoperabilidade que facilitem a integração e um desenvolvimento colaborativo. As soluções que interagem com laboratórios reais, como os de ensaios e testes elétricos, normalmente utilizam o “Labview” ficando assim o grande esforço na criação de um protocolo que permita o acesso remoto às funcionalidades deste artefato. Na conexão com laboratórios físicos verificou-se ainda muitos recursos alocados no controle de acesso, uma vez que seu uso passa a ser compartilhado remotamente. Ao se propor um modelo de desenvolvimento de objetos de aprendizagem, sugerindo ferramentas que facilitem uma abordagem estruturada, facilidade de manutenção, flexibilidade de expansão e portabilidade acredita-se que exista aqui uma contribuição deste trabalho ao cenário atual. No próximo capítulo, realiza-se uma contextualização do uso de um modelo de desenvolvimento de software reconhecido, o UML (Unified Modeling Language), quando aplicado à criação de Objetos de Aprendizagem.

Capítulo 3

MODELANDO O NÚCLEO COM A UML

3.1 A Busca pela Sistematização

Com o objetivo de definir métodos e procedimentos para sistematizar o desenvolvimento de programas de computadores (*softwares*), surgiu a Engenharia de *Software*. Estes estudos iniciados na conferência da OTAN em 1968, buscaram incorporar ao processo de fabricação de *software* as práticas e disciplinas até então utilizadas nas engenharias civil, elétrica e mecânica. Dentre os objetivos desta engenharia pode-se destacar os de se obter *softwares* mais confiáveis e cujos desenvolvimento e manutenção fossem economicamente controlados. Com a ênfase dada ao tema Qualidade, os métodos e procedimentos para o planejamento eficiente, elaboração e testes de programas de computador passaram a ser fundamentais.

A busca pela qualidade do *software* tornou-se então um dos grandes desafios da Engenharia de *software*. Não obstante, vários métodos e processos já existiam com este objetivo, pode então a Engenharia de *software* buscar a aplicação dos métodos de melhoria contínua da Qualidade com a sistematização das fases de criação, desenvolvimento e testes.

3.2 O Método PDCA

O método PDCA (*Plan, Do, Check, Act*), de melhoria contínua da qualidade, teve Deming como seu grande defensor e tem como base o controle de processos. Desenvolvido na década de 30 pelo americano Shewhart, foi aplicado nos conceitos de qualidade utilizados no Japão, onde, dando ênfase à análise e medição dos processos para a manutenção e melhoria dos mesmos, contemplou inclusive o planejamento, a padronização e a documentação.

A Sistematização buscando atingir as metas estabelecidas, é composta então de quatro fases básicas:

1. P (Plan) Planejamento: definição das metas e determinação dos métodos para alcançar as metas;
2. D (Do) Execução: educação, treinamento e execução do trabalho;
3. C (Check) Verificação: verificação dos efeitos (resultados) do trabalho executado;
4. A (Act) Ação Corretiva: atuação no processo em função dos resultados.

De maneira, um pouco, mais detalhada pode-se acrescentar:

- *Plan – Planejamento*: Definir o que se quer, planejar o que será feito, estabelecer metas e definir métodos que permitirão atingir as metas propostas. No caso de desenvolvimento de um Sistema de Informação, esta atividade pode corresponder ao planejamento do Sistema.
- *Do – Execução*: Tomar iniciativa, educar, treinar, implementar, executar o planejado conforme as metas e métodos definidos. No caso de desenvolvimento de um Sistema de Informação, esta atividade pode corresponder ao desenvolvimento e uso do sistema.
- *Check – Verificação*: Verificar os resultados que se está obtendo, verificar continuamente os trabalhos para certificar-se que estão sendo executados conforme planejados. No caso de desenvolvimento de um Sistema de Informação, esta atividade pode corresponder aos testes, análise das informações geradas e avaliação de qualidade do sistema.
- *Action – Ação Corretiva*: Fazer correções de rotas se for necessário, tomar ações corretivas ou de melhoria, caso tenha sido constatada na fase anterior

a necessidade de corrigir ou melhorar processos. No caso de desenvolvimento de um Sistema de Informação, esta atividade pode corresponder aos ajustes, implementações e continuidade do sistema.

3.3 Normas ISO/IEC

Com o objetivo de trazer uma perspectiva padronizada para tratar do assunto de qualidade de *software* [ISO98b, ISO98c], a ISO (Organização Internacional para Padronização) em seu comitê de *software* [ISO07b], elaborou a série de normas ISO/IEC 9126 [ISO91] que tratam do modelo de qualidade de produto de *software* e do processo de avaliação, respectivamente. Mais tarde, com preocupações de caráter ergonômico, ou seja, da adaptação dos meios de produção ao homem, o comitê de ergonomia da ISO elaborou a série de normas de ergonomia de *software* ISO 9241 [ISO93]. Esta série tem por finalidade auxiliar os desenvolvedores na elaboração de sistemas de *software* para Trabalho de escritório com computadores VDT (Visual Display Terminals) de maneira saudável, segura, eficiente e confortável aos usuários de computadores. Uma breve descrição destas Normas são apresentadas a seguir.

Norma ISO/IEC 9126-1

A ISO/IEC 9126-1 – Modelo de Qualidade é a primeira parte da ISO/IEC 9126 – Tecnologia da Informação/Qualidade de Produto de *software*. Esta parte da ISO/IEC 9126 [ISO00] descreve o modelo para qualidade de produto de *software* dividido em duas partes: qualidade interna e externa e qualidade em uso.

Qualidade interna e externa especifica seis características que são subdivididas em sub-características, como apresentado:

1. *Funcionalidade* é a capacidade do produto de *software* para satisfazer necessidades implícitas quando o *software* é usado sob condições específicas. Esta característica está dividida em 5 sub-características:

Adequação,

Acurácia,

Interoperabilidade,

Conformidade e

Segurança de acesso.

2. *Confiabilidade* é a capacidade do *software* de manter seu nível de desempenho quando usado sob condições estabelecidas. Está dividida em três sub-características:

Maturidade,
Tolerância à falhas e
Recuperabilidade.

3. *Usabilidade* é a capacidade do produto de *software* ser entendido, lido, usado e atrativo ao usuário quando usado sob condições estabelecidas. Está dividida em quatro sub-características:

Intelegibilidade,
Apreensibilidade,
Operacionalidade,
Atratividade.

4. *Eficiência* é a capacidade do produto de *software* para produzir desempenho apropriado, relativo a quantidade de recursos usados, sob condições estabelecidas. Está dividida em três sub-características:

Comportamento com relação ao tempo,
Comportamento com relação ao uso de recursos e
utilização.

5. *Manutenibilidade* é a capacidade do produto de *software* ser modificado. Está dividida em quatro sub-características:

Analisabilidade,
Modificabilidade,
Estabilidade,
Testabilidade.

6. *Portabilidade* é a capacidade do produto de *software* ser transferido de um ambiente para outro. Está dividida em quatro sub-características:

Adaptabilidade,
Facilidade de Instalação,
Capacidade para co-existir e
Facilidade para substituir.

7. *Qualidade* em Uso especifica quatro características, são elas:

Efetividade,
Produtividade,
Segurança e
Satisfação.

Com a necessidade de garantir ao usuário um *software* de qualidade, os aspectos de usabilidade do produto são de suma importância. Do ponto de vista do produto, a usabilidade é definida na ISO/IEC 9126 como a capacidade do produto de *software* ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições especificadas. A qualidade sob a ótica do usuário é traduzida pela qualidade em uso, cuja definição na ISO/IEC 9126 é a capacidade do produto de *software* de permitir que usuários atinjam metas estabelecidas com eficácia, produtividade, segurança e satisfação em determinados contextos de uso.

Norma ISO 9241

A ISO 9241 trata dos Requisitos Ergonômicos para Trabalho de escritório com computadores VDT (*Visual Display Terminals*). É composta por 17 partes e tem por objetivo promover a saúde e a segurança de usuários de computadores e garantir que eles possam operar estes equipamentos com eficiência e conforto.

Em uma visão mais orientada ao usuário, são definidas as características da qualidade em uso (eficácia, produtividade, segurança e satisfação).

A ISO/IEC 9241 fornece princípios ergonômicos, formulados em termos gerais, apresentados sem referências às situações de uso, aplicação, ambiente ou tecnologia e sim relacionados à apresentação da informação.

Ela está focada na organização da informação e na utilização de técnicas de codificação, com o objetivo de garantir a satisfação e o desempenho do usuário.

A seguir será apresentada uma breve descrição das partes 1, 10 e 11, que compõem a ISO 9241, com aspectos importantes para a elaboração deste trabalho.

Norma ISO 9241-1

A ISO 9241-1: Introdução Geral, apresenta uma visão geral de todas as demais partes individualmente, além do inter-relacionamento entre elas e qual o usuário esperado para cada parte [ISO97].

Norma ISO 9241-10

A ISO 9241-10: Princípios de Diálogo, fornece princípios ergonômicos apresentados sem referências a situações de uso, aplicação, ambiente ou tecnologia e estes princípios podem ser usados em especificações, projetos e avaliações de diálogos para

terminais [ISO93].

As definições aplicadas para esta parte da ISO 9241 são duas: diálogo e usuário.

- *Diálogo* é a interação entre um usuário e um sistema para alcançar um objetivo particular e
- *Usuário* que é o indivíduo interagindo com o sistema.

São identificados sete princípios considerados importantes no projeto e avaliação de usabilidade de diálogos:

1. Adequação à tarefa,
2. Auto descrição,
3. Controlabilidade,
4. Conformidade com as expectativas do usuário,
5. Tolerância a falhas,
6. Adequação à individualização e
7. Adequação ao aprendizado.

Norma ISO 9241–11

A ISO 9241–11: Usabilidade dos Sistemas tem como escopo definir usabilidade, explicar como identificar informações necessárias que se deve levar em conta para especificar ou avaliar usabilidade de um produto e fornecer um guia de um documento que contém as informações necessárias para especificação ou avaliação de usabilidade de um produto [ISO98a].

Segundo esta parte, para se medir o grau de usabilidade algumas informações são necessárias, entre elas:

- Descrição do objetivo de uso do produto,
- Contexto de uso do produto (descrição dos usuários, das tarefas, do equipamento e do ambiente) e
- Métricas a serem utilizadas para medir a eficácia, eficiência e satisfação.

A ISO/IEC 9241–11 [ISO98a], define usabilidade como o quanto o produto pode ser utilizado por usuários para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso.

3.4 Desenvolvimento baseado em UML

O UML “*Unified Modeling Language*”, foi publicado em 1997 como um método para diagramação de projetos de desenvolvimento de *software*. Foi concebido por um consórcio de grandes personalidades na área de projeto e análise utilizando orientação a objetos. Em analogia com as demais disciplinas de engenharia, a UML pode ser comparada a métodos padrões de documentação como os diagramas esquemáticos dos engenheiros Eletro-eletrônicos, as plantas e os desenhos mecânicos para os Arquitetos e Engenheiros Mecânicos respectivamente.

O valor agregado está em organizar o projeto de uma forma que clientes/usuários, analistas, programadores e outros envolvidos no desenvolvimento do sistema possam entender e convergir para uma implementação comum.

Estes são alguns dos benefícios conseguidos com a utilização da UML:

1. O *software* é documentado e projetado de maneira profissional antes de ser codificado. Pode-se conhecer antecipadamente o que se terá implementado.
2. A reutilização de código é um das possibilidades evidentes, já que o projeto do sistema é visível antecipadamente. Poupa-se tempo, esforço e custo de desenvolvimento.
3. Falhas de lógica podem ser verificadas e corrigidas a partir da visualização do sistema, reduzindo-se o número de surpresas indesejadas.
4. O projeto irá determinar como será feito o desenvolvimento do *software*. Decisões certas podem ser tomadas antes do início da codificação do *software*, o que representa menor custo de desenvolvimento.
5. Pode-se ter uma visão mais abrangente do sistema dimensionando, com isso, os recursos de “*hardware*” com maior eficiência.
6. A documentação gerada pela UML torna bem mais fácil a tarefa de alterar o sistema futuramente, necessitando de menos esforço e menor custo de manutenção.
7. O envolvimento de outros profissionais mesmo durante o processo de implementação é bem mais fácil, já que existe uma linguagem e uma documentação padronizada.

Com o uso do “*Unified Modeling Language*” é possível produzir *softwares* mais confiáveis e eficientes com um menor custo total de desenvolvimento e manutenção,

além de melhorar a comunicação entre todas as partes envolvidas. Um *software* documentado com UML pode ser modificado muito mais facilmente.

Antes de se descrever as ações de cada etapa da “UML”, conheceremos alguns conceitos importantes e suas notações quando utilizados neste modelo.

Classes: Uma classe, em linguagens orientadas a objeto, é a possibilidade de combinar num único registro, campos de dados e funções para operar os dados do registro.

Em outras palavras, as classes são os blocos de construções mais importantes de qualquer sistema orientado a objetos. Uma classe é uma descrição de um conjunto de objetos (elementos do sistema) que compartilham os mesmos atributos, operações, relacionamentos e semântica. Podem ser implementadas em uma ou mais interfaces.

Todos os objetos são instâncias de classes, onde a classe descreve as propriedades e comportamentos daquele objeto.

Em UML as classes são representadas por um retângulo dividido em três compartimentos:

Nome : Apenas define o nome da classe modelada, todas as classes devem ter um nome que as diferencie das demais. Usa-se uma seqüência de caracteres para identificá-las. Uma classe pode ter um nome simples, ou com caminho. O caminho identifica o pacote ao qual a classe pertence.

Atributos : Relação de atributos que a classe possui em sua estrutura interna, cada atributo é uma propriedade de uma classe, que descreve um intervalo de valores que as instâncias da propriedade podem apresentar. Uma classe pode ter qualquer número de atributos ou nenhum. Um atributo representa o tipo de dados ou estados que cada item de uma classe pode assumir, são representados logo após o nome da classe.

Operações : São os métodos de manipulação de dados e de comunicação de uma classe com outras do sistema. São os processos que a classe pode realizar. As operações correspondem claramente a métodos em uma classe. No nível de especificação, as operações correspondem a métodos públicos. Normalmente, não

se mostram as operações que simplesmente manipulam atributos, porque elas podem ser freqüentemente inferidas. Entretanto, pode ser preciso identificar se um dado atributo é somente para leitura (isto é, o seu valor nunca muda). No modelo de implementação, também pode-se querer mostrar operações privativas (*private*) e protegidas (*protected*).

A sintaxe usada em cada um destes compartimentos é independente de qualquer linguagem de programação, embora outras sintaxes possam ser usadas, como a do *Flash*, *C++* e *Java*.

Existem outras particularidades ligadas às classes em UML, mas não serão descritas neste trabalho.

Uma classe pode ter várias operações ou até mesmo nenhuma, que são representadas logo após os atributos.

Em UML durante a elaboração dos modelos de análise e projeto, deve-se evitar traduzi-los em códigos, cabendo esta tarefa à fase de programação.

A seguir, estão descritos os objetos, que são as classes em ação.

Os objetos são elementos que se pode manipular, acompanhar seu comportamento, criar, interagir com ele, ou até destruí-lo. Um objeto pode existir no mundo real ou pode ser uma derivação de estudos da estrutura e comportamento de outros objetos do mundo real. Corresponde a qualquer coisa que tenha algum significado para uma dada aplicação. Por exemplo, uma máquina, uma organização, ou negócio.

Um objeto é simplesmente alguma coisa que faz sentido no contexto de uma aplicação. Objeto é uma entidade independente, assíncrona e concorrente, armazena dados, encapsula serviços, troca mensagens com outros objetos, e é modelado para executar as funções finais do sistema.

Abstração de um Objeto Abstração é o princípio de ignorar os aspectos de um assunto não relevante para o propósito em questão, tornando possível uma concentração maior nos assuntos principais. Consiste na seleção que o analista faz de alguns aspectos, ignorando outros. Existem duas formas de abstração: de Procedimentos e de Dados.

Abstração de Procedimentos

Princípio de que qualquer operação com um efeito bem definido pode ser

tratada por seus usuários como uma entidade única, mesmo que a operação seja realmente conseguida através de alguma seqüência de operações de nível mais baixo.

Abstração de dados

Consiste em definir um tipo de dado conforme as operações aplicáveis aos objetos deste tipo. Porém, estes objetos só podem ser modificados e observados através destas operações.

Encapsulamento Encapsular é omitir informações pelo princípio de que uma determinada entidade esconde informações que são necessárias apenas a si mesma. É fundamental que o objeto proteja seus dados, não permitindo que o usuário do objeto os acesse diretamente, mas através de métodos, se houver necessidade.

Polimorfismo É o conceito usado em linguagens de programação orientada a objetos para denotar a característica de que a linguagem suporta a utilização do mesmo identificador (o mesmo nome) para métodos de classes diferentes.

Um conceito em teoria de tipo no qual um nome (como uma declaração de variável) pode denotar objetos de muitas subclasses diferentes que são relacionadas por alguma superclasse comum, assim, qualquer objeto denotado por esse nome tem a capacidade de responder a algum conjunto comum de operações de modos diferentes.

Objetos na UML Em UML um objeto é mostrado como uma classe, só que seu nome é sublinhado e o nome do objeto pode ser mostrado, opcionalmente, precedido do nome da classe.

Precusores da análise orientada a objeto, defendiam que programas de computador deveriam ser estruturados de acordo com o problema a ser resolvido. O termo Orientação a Objetos sugere abstrações do mundo real e trechos de programas de computador, ou objeto.

Reusabilidade Um grande fator da orientação a objetos é a reusabilidade. Reutiliza-se códigos de programas desde o início da computação. As técnicas de orientação a objetos permitem muito mais que a reutilização de códigos; pode-se reutilizar requisitos, análise, projeto, planejamento de testes, interfaces de usuário e arquiteturas, ou seja, todos os componentes de engenharia de *software* podem ser encapsulados como reutilizáveis.

3.4.1 Levantamento de Requisitos

Saber para onde se quer ir é um bom começo para qualquer caminhada. O levantamento de requisitos tem a função de capturar as intenções e necessidades dos usuários do *software* a ser desenvolvido.

Em resumo pode-se relacionar as seguintes etapas principais desta fase:

Descobrir a linguagem e o negócio do usuário (cliente): Aqui o analista ganha entendimento sobre os processos na linguagem e do ponto de vista do usuário, entrevistando seu cliente e conhecendo o(s) processo(s) passo a passo. Produto: Diagrama de Atividades.

Executa o domínio da Análise: A partir da entrevista o analista ganha conhecimento das principais entidades no domínio do cliente. São identificados os nomes que serão as classes e outros atributos. Os verbos normalmente são associados às operações das classes. Produto: Diagrama de Classe.

Identifica os Sistemas Cooperativos: No início do processo os desenvolvedores identificam quais as dependências do novo sistema com o sistema atual. Produto: Diagrama de Execução.

Verificação dos requisitos: Executa-se neste ponto a primeira reunião conhecida como JAD (Joint Application Development), com a participação dos patrocinadores (decision makers), os usuários potenciais (potential users) e os membros da equipe de desenvolvimento. Um facilitador atua como moderador e solicita dos usuários e patrocinadores o depoimento do que eles desejam do futuro sistema. As declarações são registradas e os modeladores de sistema refinam o diagrama de classes gerado anteriormente.

Apresentação dos Resultados: Uma vez finalizada a fase de levantamento o Gerente de Projeto apresenta e convalida os resultados com o cliente.

Uma ferramenta fundamental neste modelo, e que é utilizada em vários momentos, são as funções chamadas “casos de uso” (“*use-cases*”), que descrevem as necessidades e as operações baseadas em situações reais.

Na análise de requisitos identifica-se, documenta-se o que é realmente necessário, e comunica-se a todos os envolvidos no projeto, de forma mais clara possível e sem ambiguidades buscando também identificar os riscos envolvidos.

Através das funções “casos de uso”, figura 3.1, modelam-se os “atores externos” (“*actors*”), que são entidades (usuários ou outros sistemas externos) que interagem

com o sistema a ser desenvolvido. Os atores externos e os “casos de uso” são modelados com relacionamentos que possuem comunicação associativa entre eles ou são desmembrados em hierarquia. Descreve-se um “caso de uso” através de um texto especificando os requisitos do ator externo que utilizará este “casos de uso” . Um “caso de uso” tanto pode depender de um ou mais “casos de uso” como pode ter seus dependentes.

Utilizando-se diagrama de “casos de uso” pode-se mostrar aos futuros usuários o que se pode esperar do sistema em desenvolvimento.

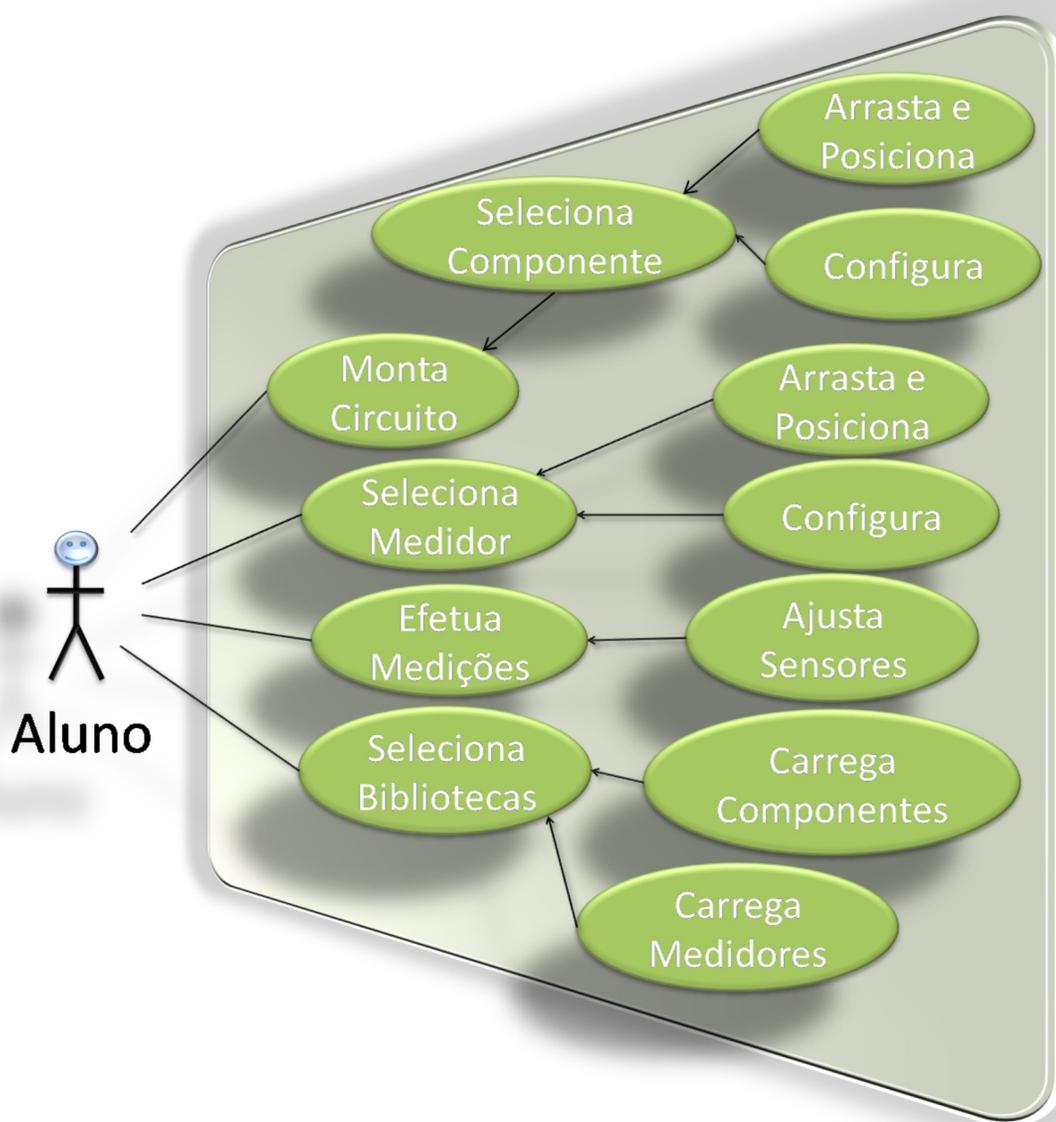


Fig. 3.1: UML – Diagrama Casos de Uso
Apresentação parcial do diagrama “Caso de Uso” para implementação do Laboratório Virtual, conforme modelo UML.

Levantamento e análise de requisitos do Laboratório Virtual

Entrevista com os alunos e docentes e com as pessoas que apóiam o curso onde será utilizada a ferramenta de *software* educacional.

Definição dos objetos do sistema : aluno de Engenharia, professor de Engenharia, fonte de tensão, resistor, capacitor, indutor, motor, terminal, analisador, bancada Elétrica, sinalizador, alarme, gerador de Onda etc..

Definição dos atores do sistema : Aluno, Educador, Sinais de Entrada e Sinais de Saída.

Definição e detalhamento dos casos de uso: Preparar ambiente, montar circuito, efetuar medições.

Definição das classes : Aluno, Educador, Componente, Terminal, Bancada, Biblioteca de Componente, Medidor, Biblioteca de Medidor, Painel de controle, Dispositivo de Saída, Dispositivo de Entrada etc., veja figura 3.2.

Definir os atributos : Nome, Tipo, Curso; Resistência, Temperatura, Pressão, Tensão, Corrente, Campo Magnético, Torque, Velocidade, Direção, Peso; Número de Canais, Escala de Amplitude, Escala de Tempo, Diagrama fasorial...

Definir métodos das classes : *selecionarBiblioteca, selecionarComponente, configurarComponente, calcularFuncaoMatematica, posicionarComponente, selecionarMedidor, posicionarMedidor, configurarMedidor, posicionarSensor...*

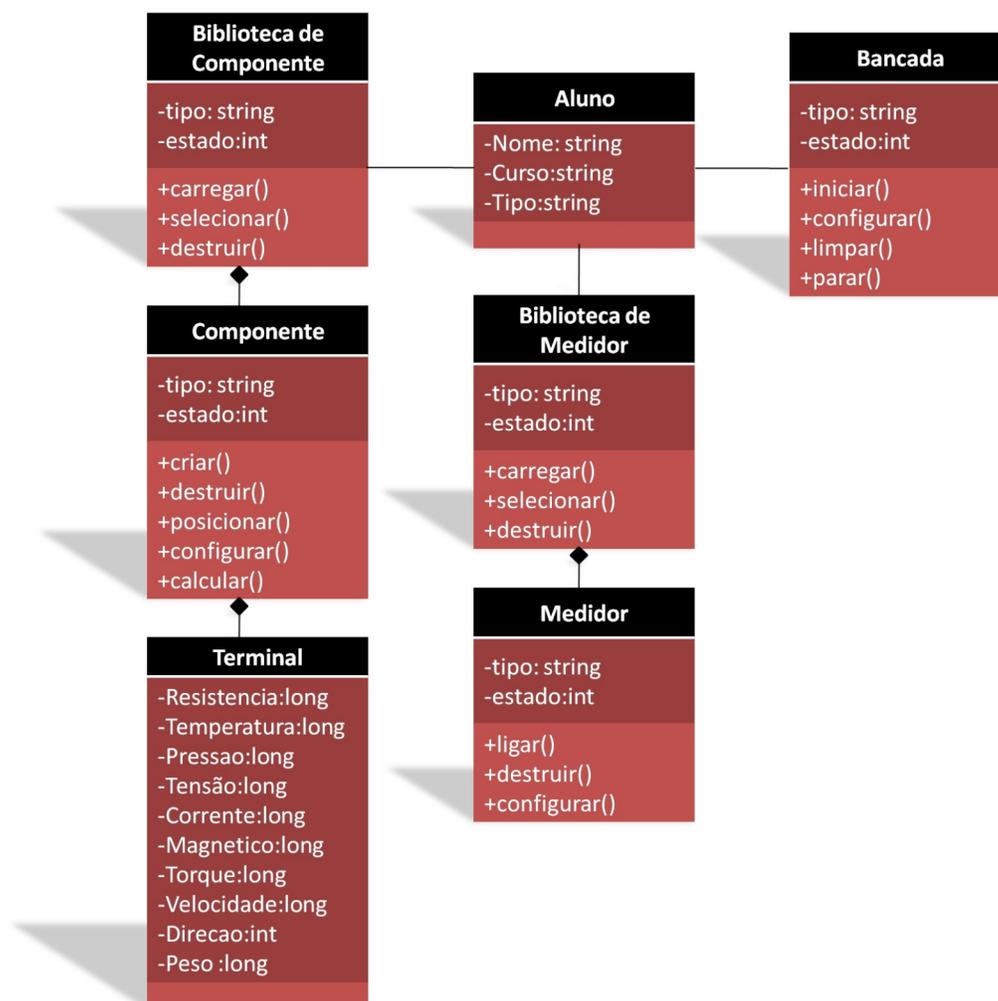


Fig. 3.2: UML – Diagrama de Classes

Apresentação parcial do diagrama de “Classes” para implementação do Laboratório Virtual, conforme modelo UML.

Como resultado desta fase do projeto pode-se identificar algumas necessidades importantes a serem consideradas durante a implementação:

1. Tempo de resposta, onde a velocidade de carga da ferramenta, das bibliotecas e das interações com os estudantes tem que ser rápida pois é preciso apenas poucos segundos para o usuário perder o foco no diálogo;
2. Ícones, que permitam o acesso rápido às sequências de comandos ou serviços do sistema;
3. Estilo das telas, utilizando-se sempre um mesmo estilo de tela em todo o sistema permitindo apenas pequenas alterações para conteúdos muito diferentes, mantendo assim um padrão de apresentação;
4. Impactos da utilização do sistema sendo carregado como um *frame* de outra plataforma de *software* educacional, como o “Teleduc” por exemplo. O dimensionamento eficiente das imagens e visibilidade dos recursos precisam ser preservados.
5. A escrita, principalmente em aplicações educacionais, deve ser utilizada para apresentar um conteúdo sucinto, parágrafos pequenos e listas com *bullets* que além do conteúdo deve possuir como diferencial uma gramática correta;
6. Estrutura de apresentação do conteúdo da maneira mais atraente possível, utilizar animação listas com *bullets* para quebras de textos uniformes, além de explorar o hipertexto, sem contudo comprometer a flexibilidade e a velocidade mínima de interação;
7. Linguagem clara, pois usuários normalmente lêem somente a primeira sentença de cada parágrafo, portanto utilizar frases conclusivas para que o aluno ao ler a primeira sentença entenda exatamente o que a tela apresenta, além de apresentar textos de tamanho adequado e de fácil leitura;
8. Legibilidade com relação a cores, contraste de cor de letra com cor de fundo, tamanho e tipo das letras utilizadas;
9. Documentação *on-line* também é um item a ser utilizado pelos alunos, que normalmente só irão procurar por documentação em caso de necessidade de informação ou dificuldade encontrada. Deve ser incluído neste item o glossário, um passo-a-passo e abundância de exemplos buscando auxiliar ao máximo o usuário em suas necessidades;

10. Multimídia é uma boa opção a ser utilizada para obter a atenção do usuário mas deve ser utilizada com muita cautela para não sobrecarregar o usuário com excesso de efeitos de animação, vídeo e áudio;
11. Tempo de execução, em caso de processos lentos é recomendado que seja informado ao usuário quanto tempo levará para ser concluído;

3.4.2 Análise

A Análise Orientada a Objetos, conceitualmente é o estudo de um problema, antes de qualquer ação. É o estudo do domínio de um problema que leva a uma especificação de comportamentos observáveis externamente. É uma declaração completa, consistente e possível do que é necessário, um conjunto de características operacionais quantificadas e funcionais.

Analisar é obter as necessidades de um sistema e o que precisa ser desenvolvido para satisfazer as necessidades do usuário. Analisar não é definir como o sistema será desenvolvido, mas sim investigar o problema.

A AOO (Análise Orientada a Objetos) tem dois propósitos. Primeiramente formalizar uma “visão” do mundo real dentro do qual o sistema será desenvolvido, estabelecendo os objetos que servirão como principais estruturas organizacionais do sistema de *software* e também as que o mundo real demanda. Em segundo lugar, a AOO formaliza a colaboração de um dado conjunto de objetos na execução do trabalho do sistema de *software* que está sendo especificado. Esta formalização representa como cada objeto se comunica com os demais.

O Projeto Orientado a Objetos (PjOO) é visto como o processo de especificação das partes da construção, ou seja, instruções, guias, recomendações, estipulações, qualificações, regras, etc.. Utilizam-se estes processos para implementar um sistema em um ambiente específico, em busca da solução do problema.

Durante a PjOO é dada ênfase na elaboração dos elementos lógicos do *software*, sendo desenvolvidos utilizando-se uma linguagem de programação orientada a objetos, os quais possuem atributos e métodos.

A fase de análise preocupa-se com as primeiras abstrações (classes e objetos) e

mecanismos presentes no contexto do problema. Conforme apresentado anteriormente descreve-se as classes nos Diagramas de Classes e também para ajudar na descrição dos “*use-cases*”, podendo estas estarem ligadas uma nas outras através de relacionamentos.

Pode-se dividir esta fase nas seguintes ações:

Entendimento do sistema: Em uma reunião “*JAD*” (*Joint Application Design*) com os usuários, o time de desenvolvimento trabalha para identificar os “atores” que iniciam cada “caso de uso” em estudo. Identificam também os atores que interagem com os demais, lembrando que um ator pode ser uma pessoa ou mesmo outro sistema.

Produto: Diagramas “Caso de Uso”.

Verificação dos “Casos de Uso”: Ainda em reunião com os usuários procura-se detalhar e verificar a sequência de passos na execução de cada “caso de uso”.

Produto: Descrição detalhada de passos em cada “Diagrama de Caso de Uso”.

Refinamento dos Diagramas de Classe: O modelador de objetos faz o refinamento do diagrama de classes durante as reuniões “*JAD*”, identificando as associações, as classes abstratas, as multiplicidades, generalizações e as agregações.

Produto: Diagrama de Classes Refinada.

Análise de Mudança de Estado dos Objetos: O modelador de objetos faz o refinamento do modelo mostrando as “Mudanças de Estado” onde forem necessárias.

Produto: Diagrama de Estado.

Define as interações entre os objetos: O time de desenvolvimento define como os objetos interagem a partir do conjunto de “Casos de Uso” e os “Diagramas de Classes”.

Produto: Diagramas de Sequência e de Colaboração.

Análise da Integração entre Sistemas Cooperativos: O Engenheiro de Sistemas, trabalhando em paralelo com as ações anteriores, identifica detalhes específicos da integração entre sistemas cooperativos. Por exemplo:

Comunicação: Que tipo de comunicação está envolvida?

Arquitetura de Rede: Qual a arquitetura de rede utilizada?

Banco de Dados: Se o sistema utiliza algum banco de dados? e de que tipo?

Produto: “Diagrama de Execução” detalhado e se necessário o “Modelamento de Dados”.

3.4.3 Projeto

O Projeto cria uma representação do domínio do problema do mundo real e leva-a a um domínio de solução que é o *software*.

Nesta fase parte-se para as soluções técnicas, através dos resultados obtidos na fase da análise. Serão adicionadas novas classes para oferecer uma infra-estrutura técnica tais como: interface do usuário e periféricos, banco de dados, interação com outros sistemas, e outras mais. É feita uma junção das classes da fase da análise com as classes técnicas da nova infra-estrutura, podendo assim alterar tanto o domínio principal do problema quanto a infra-estrutura.

Com a elaboração do projeto obtém-se detalhadamente as especificações (requisitos), para dar início a fase de programação.

Tendo sempre em mente que o importante é entender o Usuário, consideram-se as seguintes etapas:

Desenvolver e refinar os diagramas de objetos: Programadores utilizam os diagramas de classe para identificar os diagramas de objetos necessários com seus respectivos diagramas de atividades que são a base para boa parte do código a ser implementado.

Produto: Diagrama de Atividades.

Desenvolver o Diagrama de Componentes: Busca-se as dependências entre os componentes do sistema.

Produto: Diagrama de Componentes.

Planejamento de Execução: Com base no diagrama de Componentes planeja-se a execução e a integração com outros sistemas.

Produto: Parte do diagrama de execução.

Projeto de Interfaces com usuários: Uma nova JAD apresenta e ratifica as expectativas de usuários, analistas, programadores e projetistas quanto às interfaces, telas e informações que correspondam aos casos de uso.

Produto: Protótipos das telas de interface com o usuário.

Desenvolvimento de Testes: Preferencialmente alguém de fora da equipe de desenvolvimento cria as ferramentas e propõe as situações de teste baseadas nos casos de uso.

Produto: Procedimentos de Teste.

Documentação: É iniciada a documentação do sistema que deve alcançar a forma mais completa para cada documento.

Produto: Estrutura da documentação.

Usabilidade deve ser uma preocupação constante, tão importante que dedicou-se o capítulo “USABILIDADE - DESENVOLVENDO O AvalUEAD” somente para tratar deste assunto. Para preservar-se a usabilidade deve-se investir na simplicidade e na criatividade com que se constrói um sistema, devendo este apresentar um conteúdo de alta qualidade, rápida atualização, menor tempo de “*download*” e ser fácil de usar além de atender as necessidades do usuário [Nie98, Nie02a].

É importante salientar que a usabilidade de um sistema está sempre associada às características dos usuários, como:

- tarefas,
- equipamentos,
- ambientes físicos e organizacionais.

Através das avaliações de usabilidade é possível obter resultados surpreendentes, como diagnosticar as reais características do projeto e constatar problemas efetivos de utilização do sistema durante as interações.

Dentro dos ciclos de desenvolvimento, os processos de avaliação sugerem ações de “re-projeto” mais evidentes devido aos problemas de interação efetivos ou diagnosticados [Tam02].

Dos riscos mais evidentes em projetos de software, pode-se destacar pelo menos quatro que estão associados a problemas de usabilidade. São eles:

1. Mudanças freqüentes solicitadas pelo usuário,

2. Necessidades não observadas,
3. Falta de entendimento do usuário sobre seus requisitos e
4. Falta de comunicação e entendimento entre analistas e usuários.

3.4.4 Implementação

Para que uma fase de programação possa ter um bom desempenho, necessita-se de um projeto bem elaborado, conseqüentemente converter as classes da fase do projeto para o código da linguagem orientada a objetos escolhida. Se o projeto foi elaborado corretamente e com detalhes suficientes, a tarefa de codificação é facilitada. A complexidade dessa conversão vai depender da linguagem escolhida, no entanto esta tarefa pode tornar-se fácil ou difícil de se realizar.

Codificação: De posse dos Diagramas de Classes, Objetos, Atividades e Componentes, os programadores constroem o código dos programas do sistema.
Produto: O Código.

Testes de Código: Esta ação realimenta a codificação e vice-versa, até que o código tenha passado por todos os níveis de teste.
Produto: Resultados dos testes de codificação.

Construção e testes das Interfaces de Usuário: Esta ação faz a ligação entre projeto e desenvolvimento. Um especialista em interfaces (*GUI – Graphical User Interface*) constrói as telas de interação previamente aprovadas e faz sua associação com o código e testa.
Produto: Sistema completo em operação, com as interfaces de usuário.

Documentação Completa: Durante as etapas de desenvolvimento, especialistas em documentação trabalham em paralelo com os programadores para completar toda a documentação.
Produto: Sistema documentado.

3.4.5 Testes de Sistema

A etapa de testes faz parte de um ciclo de verificação e codificação até que se alcance os resultados esperados de maneira confiável.

Plano de backup e recuperação: Os engenheiros de sistema criam um plano de recuperação a ser seguido em caso de pane no sistema.

Produto: Plano de recuperação (*The crash recovery plan*).

Instalação do sistema em hardware apropriado: Nessa etapa instala-se o sistema em seu ambiente destino, com o hardware e as interações planejadas.

Produto: Implantação do Sistema Desenvolvido.

Teste do Sistema Instalado: O time de desenvolvimento executa os testes do sistema instalado nos computadores destino.

1 – Executa as funções como deveria?

2 – O sistema de recuperação funciona em caso de pane?

Os resultados destes testes determinam se novas alterações serão necessárias.

Produto: Resultado dos Testes.

3.5 Conclusão deste capítulo

Neste capítulo apresentou-se de maneira resumida, referências e recomendações para a criação de interfaces destinadas a interação entre homem e máquina. Destacou-se também a importância da sistematização no desenvolvimento de software e a contribuição da engenharia ao ser aplicada na metodologia de criação e manutenção de programas de computadores. Utilizando-se dos conceitos da UML (Unified Modeling Language) efetuou-se o modelamento de classes e objetos para a implementação do núcleo do modelo proposto neste trabalho. No próximo capítulo apresenta-se o modelo tridimensional de desenvolvimento, além da arquitetura e a implementação do núcleo do Laboratório Virtual.

Capítulo 4

O DESENVOLVIMENTO DO NÚCLEO VIRTUAL

4.1 Diagrama Esquemático Tridimensional

Uma ferramenta educacional eficiente deve permitir o desenvolvimento dos argumentos que levam à formação do conhecimento. Portanto, pode-se perguntar, porque estudar um resistor como um componente básico da eletrônica que possui as características pré definidas? Não seria interessante permitir que o estudante criasse um componente cujas especificações chegassem a ser comparadas com um componente conhecido ou, até mesmo, um novo componente?

Recentemente foi apresentado o Memristor, [HPL08] um novo componente básico para a eletro-eletrônica. Este novo componente já poderia existir em um ensaio simulado, dentro de um laboratório virtual pedagógico, para atender a uma necessidade fora dos paradigmas que congelam o desenvolvimento e até mesmo a criatividade. A necessidade poderia ser o desejo de se tratar circuitos como estados de energia, de forma que não fosse necessário carregar estes estados de uma memória secundária a cada parada do sistema. Se assim fosse não seria necessário recarregar o sistema operacional de um computador toda vez que se desliga e religa aquele equipamento. Situação semelhante ao processo de se dirigir um carro onde se fornece energia ao sistema para movimentá-lo e depois se fornece energia novamente para perder o movi-

mento quando se deseja pará-lo. Porque não conservar o nível de energia adquirido através de um processo de transformação que possa restaurá-la? São questões difíceis de serem respondidas sem a abstração de algumas algemas tecnológicas ou paradigmas.

Como inovação, o Memristor deve promover uma partida com muito maior velocidade às máquinas eletrônicas. Ele também se propõe a libertar as memórias da necessidade de serem permanentemente alimentadas por energia para sua conservação. Com os Memristores será possível desenvolver sistemas muito mais eficientes, pois essas novas memórias serão capazes de reter a informação mesmo quando o equipamento estiver desligado. O Memristor preserva as características da corrente, sua intensidade, e por quanto tempo circulou, mesmo quando desligado o sistema.

A inovação já propôs várias mudanças como a criação da microeletrônica que nasceu praticamente com o transistor, dispositivo revolucionário inventado em 1947, por três cientistas dos Laboratórios Bell (William Shockley, John Bardeen e Walter Brattain), para substituir as velhas válvulas a vácuo. Por sua invenção, esses três cientistas ganharam o Prêmio Nobel de Física de 1956.

Em 1958, surgia o primeiro circuito integrado, componente que reunia diodos, resistores, capacitores, transistores e indutores sobre uma pastilha de silício. Em quatro décadas, o processo de miniaturização dos componentes eletrônicos tem evoluído de tal forma que já permite reunir mais de um bilhão de transistores num único circuito integrado ou chip.

O mais extraordinário é que, ao longo dos últimos 50 anos, a microeletrônica se tornou a mais poderosa alavanca de transformação do mundo. De fato, seria muito difícil imaginar o mundo atual sem a microeletrônica.

Construir sistemas que permitam a evolução de idéias, a diagramação de modelos novos e conhecidos são alguns dos objetivos do projeto de um Laboratório Virtual. Um desenvolvimento que parte de um ambiente teórico, uma base de informações, até sua execução física, figura 4.1, em um modelo organizado em níveis ou camadas. Propõe-se dividir o projeto, separando as dificuldades e as atribuições de cada etapa, ainda segundo a proposta triádica de Charles Sanders Peirce [Not95] intensificando o processo de cognição na mente do estudante, além de promover uma melhor integração e utilização do sistema.

Cada camada é responsável por garantir acessibilidade e a estimulação associada com a Primeiridade, a Secundidade e a Terceiridade de Peirce respectivamente, considerando ainda flexibilidade, facilidade e agilidade na utilização dos recursos.

A partir da base de informações até sua execução física existem os níveis necessários



Fig. 4.1: O sistema em camadas

para organizar as tarefas em uma pilha de atribuições.

Cada nível tem sua atribuição com sua proposta de *software e hardware*, se necessários.

4.2 Descrição das Camadas

No Nível Teórico é onde se encontra o conjunto de informações com o objetivo de suscitar as lições e também as situações de teste, definições, processos da aprendizagem, características dos sistemas de monitoramento, tipos de dispositivos e sinais de entrada e saída, descrição e conceitos associados a cada tipo de componente, toda a teoria da Engenharia Elétrica abordada nos cursos, o comportamento dos sinais e dos equipamentos. Um arcabouço de dados vagos que devem ser expostos aos estudantes em busca da Primeiridade proposta por Peirce.

O Diagrama é uma proposição relativa ao esquema do sistema elétrico a ser analisado. Uma associação de idéias em busca da Secundidade de Peirce. Definido em blocos, associa as funcionalidades principais dos circuitos, os equipamentos de monitoração, fontes de alimentação (identificando cada formato de sinal elétrico a ser utilizado) e as possibilidades de entrada e saída.

Os demais níveis devem propor a Terceiridade segundo a Semiótica de Peirce. Neste, o Núcleo do Sistema é projetado para ser capaz de trabalhar de maneira independente de todas as outras camadas, porém contendo as duas anteriores (Níveis Teórico e de Diagramas). Ele é responsável pelas atividades virtuais de montagem, monitoramento e interação com o usuário, as ações necessárias ao entendimento e a formação do conhecimento. Os níveis seguintes só serão utilizados na necessidade de se efetuar uma implementação física, remota ou não.

O Nível Cliente é a interface utilizada para interagir com o núcleo do modelo e também para enviar comandos para o servidor em uma configuração remota, se implementada.

O Nível Servidor é uma interface intermediária remota utilizada, na maioria das vezes, para interagir com a implementação física captando sinais e enviando-os para as camadas superiores, ou enviando comandos para operar o circuito real de acordo com os controles recebidos da camada Cliente, em uma estrutura cliente/servidor.

O Nível Físico é o circuito físico que está sendo testado ou monitorado pelos níveis superiores da implementação 3D.

A análise a partir das camadas superiores inicia-se com a base de informações teóricas e assim por diante, figura 4.2. Não obstante, o foco será a descrição do Núcleo do Sistema Virtual, objeto desta dissertação.

Cada camada é responsável por garantir acessibilidade e os melhores resultados, considerando flexibilidade, facilidade e agilidade na utilização dos recursos.

O desenvolvimento de programas de computadores para uso educacional deve considerar ferramentas que permitam atender as necessidades específicas deste tipo de ambiente, como tempo de resposta às interações, animação e apresentação. Além disso, sua criação deve seguir um processo sistematizado que permita a visualização dos “objetos”, “métodos” e “processos”. Precisa também atender normas pertinentes como ISO 9241 [ISO97, ISO93] e ISO/IEC 9126 [ISO00], além de considerar conceitos cognitivos [Lar87, Cas05, Mar07] e ser submetido a testes de Usabilidade [Nie02a, Nie95], dentre outros [Pag04, Ngu01, Oli05].

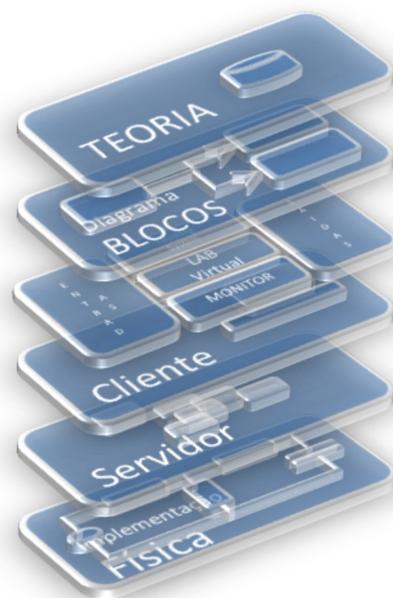


Fig. 4.2: Pilha Tridimensional de Implementação

4.3 O Núcleo do Sistema Semiótico

4.3.1 A Estrutura do Núcleo

Todos os módulos necessários para a criação de um ambiente virtual destinado ao aprendizado, incluindo recursos de monitoramento e sistemas educativos devem ser encontrados no núcleo do sistema virtual. Existem dispositivos de entrada e saída para ajustar sinais especiais e demonstrar os efeitos em alarmes, painéis sinalizadores ou simplesmente em pontos de medição. No desenvolvimento, com o objetivo de obter-se velocidade, flexibilidade, portabilidade e a boa qualidade necessária, foi utilizado o “*FLASH*” como *software* de desenvolvimento e “Imagens vetoriais” como um eficiente formato de imagens. Além disso a utilização de signos, neste nível, associado a ícones [Not95], possibilita uma versatilidade e uma grande simplicidade na geração, configuração e movimentação de elementos dentro dos objetos de aprendizagem [Sos02].

Desde que esta proposta utiliza a estrutura de uma pilha de desenvolvimento, no núcleo a maioria dos processos simulados serão feitos sem a utilização de uma arquitetura do tipo “Cliente/Servidor”, não sendo assim limitados por uma capacidade de tráfego de rede e alocação dos recursos físicos presentes nos níveis seguintes. A conexão com o ambiente físico, utilizando circuitos não virtuais, seria feito portanto utilizando-se uma rede local ou até mesmo a Internet. Em uma modelagem 3D (*Tridimensional*), pode-se verificar no núcleo do sistema, figura 4.3, componentes virtuais das demais camadas, desde os repositórios com sua base teórica até as características encontradas na implementação física, porém todos com uma abordagem diferente. Lembrando que na teoria Semiótica de Peirce a Terceiridade inclui a Secundidade e a Primeiridade [Not95].

De maneira análoga, pode-se identificar cada componente do Núcleo como um resultado das camadas anteriores do modelo 3D proposto. Uma implementação modular pode fornecer ainda, boas alternativas para a maioria das dificuldades enfrentadas neste tipo de projeto, como velocidade de resposta e qualidade de apresentação, por exemplo. No núcleo a montagem dos componentes Virtuais é implementada utilizando-se um “PV”, que é um painel virtual, funcionando como uma célula do PBV (Protoboard Virtual), painel de configuração de circuitos virtuais. O “PV” foi criado para receber cada elemento virtual e interconectá-lo ao restante do painel de conexões. O “PV” é representado por uma pastilha, figura 4.4, que ao receber o elemento virtual e suas conexões, permite a interação deste componente com o restante do ambiente. Utilizando-se da função de varredura do PBV e das funções



Fig. 4.3: Núcleo do Sistema

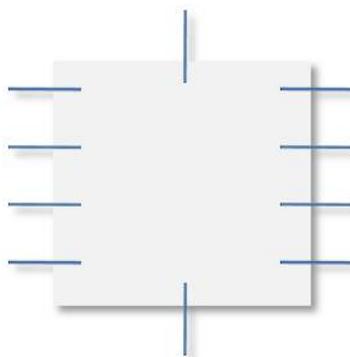


Fig. 4.4: PV10

matemáticas de cada componente, a propagação e a resposta dos sinais podem ser verificadas em cada terminal do circuito.

A busca por um painel que permitisse a simulação de elementos básicos utilizados na educação de Engenharia Elétrica sugeriu um “PV10”, figura 4.4, uma base com dez conexões distribuídas de tal forma que fosse possível montar um transistor, um resistor, um capacitor, um transformador, figura 4.8, e até mesmo um motor elétrico, dentre outros. A proposta, no entanto, é criar uma pastilha virtual que possa receber o maior número de elementos com as características desejadas.

Esta definição atende os objetivos de implementar elementos básicos da Engenharia Elétrica, com simplicidade e facilidade de configuração. Outra característica no entanto, que se mostrou de difícil desenvolvimento, foi a rotação do componente. Como pode ser visto na figura 4.5, para oferecer uma maior flexibilidade na criação, movimentação e diagramação de circuitos, um painel que permitisse uma configuração simétrica se tornou necessário, conforme figura 4.6. O PV10 foi então modificado e um painel de 12 pinos se mostrou mais adequado para montagens dos circuitos elétricos que pretendia-se utilizar. Os painéis virtuais (PV) receberão os signos que

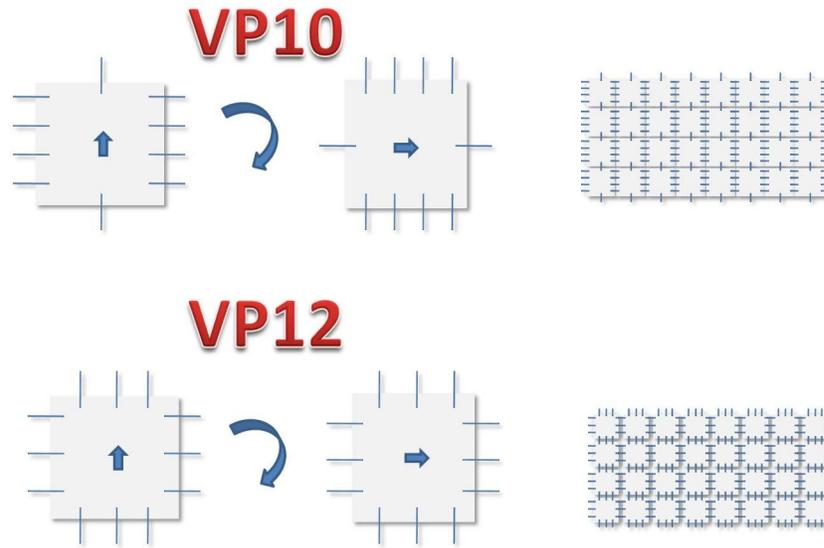


Fig. 4.5: PV10 ou PV12...

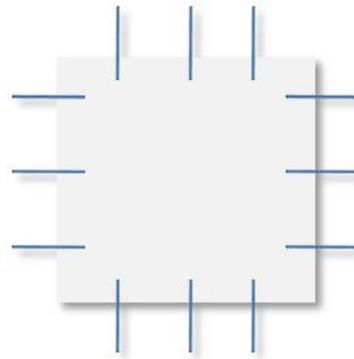


Fig. 4.6: PV12

devem associar ícones e comportamentos, expressos por fórmulas matemáticas interagindo com o restante do ambiente, o circuito, por exemplo. O resultado deve ser formado na mente do aluno promovendo o processo de cognição e a espiral do conhecimento proposta pelos estudiosos do conhecimento [Not96].

O Circuito Virtual utiliza um ou mais painéis virtuais (PV12). Quando conectados, todos os componentes virtuais vão modificar os sinais de entrada, passo a passo, em cada parte do circuito virtual e os resultados poderão ser visualizados através dos dispositivos de saída, se houverem. Enquanto isso, os sinais podem ser analisados através do sistema de monitoramento que, quando conectados, permitem a visualização ou a medição do comportamento dos sinais, figura 4.9 ou ainda, das alterações mecânicas e magnéticas em cada ponto do ambiente virtual.

O Painel Virtual é utilizado para receber cada componente necessário à montagem

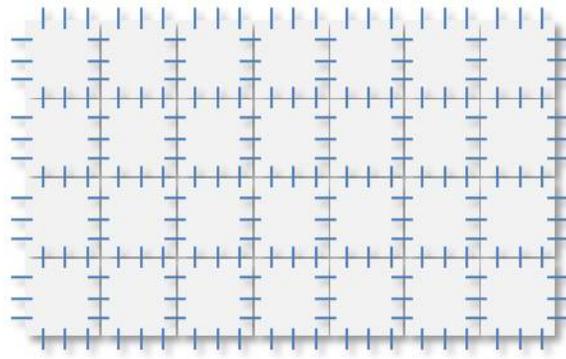


Fig. 4.7: PBV para PV12

do circuito. Este componente é um objeto virtual, como será descrito adiante, composto por um ícone e suas características com as quais ele modifica as informações no “PBV”. Inseridos em um repositório de componentes podem ser replicados, instanciados, mantendo-se as mesmas características de imagem e comportamento do componente original.

A montagem dos diversos componentes virtuais acontece sobre uma base, o “PBV” figura 4.7, cuja célula é compatível com o desenho e distribuição de um painel virtual. O campo de ensaios do Núcleo é o “PBV” (*Protoboard Virtual*). Constitui-se de uma base onde podem ser configurados vários circuitos virtuais utilizando-se, para isso, do repositório de componentes virtuais.

O PBV(*Protoboard Virtual*) é então o ambiente criado para se montar circuitos elétricos virtuais. A matriz final com cada modelo matemático, associado a cada componente virtual, deve responder de acordo com o circuito real.

A Varredura Matemática, assim aqui definida, é o cálculo seqüencial da variação dos sinais ao longo do circuito virtual. Ela considera em cada célula do PBV a contribuição individual de cada PV12 (Painel Virtual) montado ou não, bem como o relacionamento possível com os painéis adjacentes, segundo os modelos matemáticos de cada componente virtual. O processo de varredura inicia no canto superior esquerdo e prossegue coluna por coluna, de cima para baixo, até o canto inferior direito do PBV. No entanto, este processo de varredura pode ser mais complexo, respeitando as leis ou as regras que regem o ambiente (o circuito neste caso). Fontes de sinal, por exemplo, podem promover um sistema de varredura secundária recursiva mapeando as malhas de circuito e sua possível interação com outras fontes, se houver. Um processo que é natural para a acomodação de cargas em um circuito real, porém pode se tornar muito sofisticado ao ser emulado.

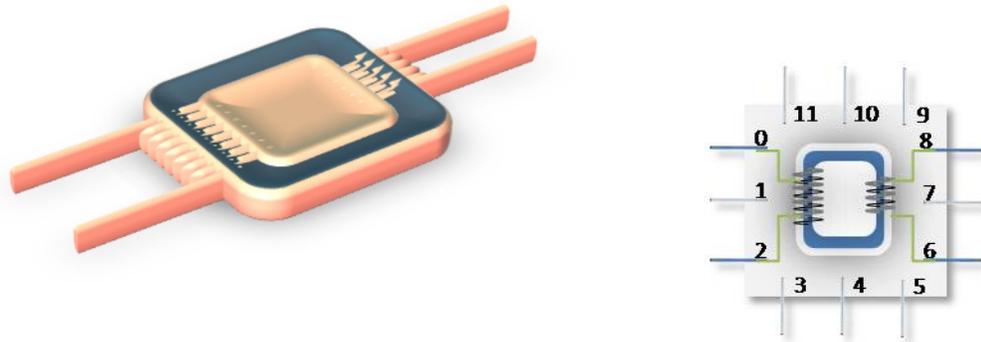


Fig. 4.8: Transformador Elétrico em um PV12

4.3.2 Representação do Painel Virtual

Cada PV12 montado terá suas equações que calculam o estado de tensão e corrente, por exemplo, em cada um dos seus doze pinos. Estes sinais serão considerados como sinais de entrada que vão interferir nos cálculos dos painéis adjacentes.

A representação do PV12 montado utiliza uma distribuição de pinos que será referenciada em equações aplicadas no modelo matemático dos respectivos componentes. Ferramentas de configuração também estão disponíveis para modificar as características do componente acoplado. Um exemplo pode ser um transformador elétrico, conforme mostrado na figura 4.8, em um esquema virtual. Este pode ser representado por um PV12 montado que utiliza os pinos 0 e 2 para os sinais de entrada e os pinos 6 e 8 como conectores de saída e os demais pinos não são utilizados. Com uma ferramenta de configuração dinâmica pode-se mudar o número de espirais no rolamento do primário, bem como no secundário. Se a razão entre as espirais for menor que um, isso significa uma voltagem menor, uma atenuação, sendo assim verifica-se uma queda de tensão na saída do transformador se comparada com o estágio primário do mesmo. O efeito pode ser visto através das telas de dois analisadores no sistema de monitoramento, figura 4.9: o primeiro conectado aos pinos 1 e 2 e o segundo analisador conectado aos pinos 6 e 8 do mesmo PV12 que representa o transformador elétrico.

No Painel Virtual se encontram os principais recursos para a definição, criação e socialização dos signos, estruturado por pelo menos um ícone, uma organização de conexões em várias dimensões e um modelo matemático que o permite interagir com o ambiente onde será utilizado.

Enquanto signos, podem ser reproduzidos e cada réplica manterá as características de comportamento originais, porém seus parâmetros podem ser alterados através do

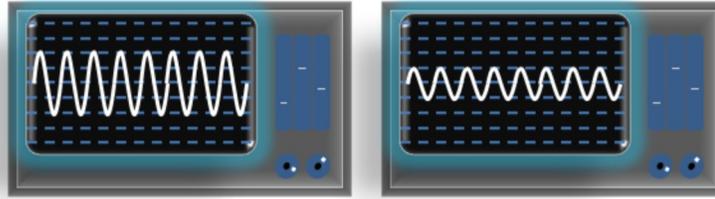


Fig. 4.9: Sinais de entrada e saída no monitor

painel de controle do Laboratório Virtual.

Por definição, um PV12 montado, que agora traz características do componente que o modifica, pode também conter outras camadas funcionais sendo que a primeira delas, de nível mais elevado, normalmente será utilizada para apresentar informações relativas à identificação do componente virtual, seu controle de navegação e de exclusão, e ainda sensível ao “mouse” que a torna visível.

Com o sistema de monitoração pode-se visualizar os efeitos do transformador elétrico. Para que fosse possível montar circuitos virtuais de teste em corrente alternada é que foi proposta a estrutura do laboratório de corrente alternada dentro do núcleo do sistema.

4.3.3 O Laboratório de Corrente Alternada

Com esta plataforma educacional pode-se propor situações de acordo com as instruções teóricas do sistema educador que poderão ser acessadas, no formato *FLASH*, por vários sistemas de apoio à educação como o *Teleduc*, *Moodle* e outros. Uma vez que se trata de uma aplicação desenvolvida em *Flash* pode ser facilmente carregada e utilizada através de um navegador popular como o *Internet Explorer*, *Mozilla* ou o recente *Chrome*, por exemplo. Pode ainda ser chamado diretamente do *TelEduc* ou *Moodle* através de um *hiperlink* para o arquivo com extensão “*swf*”. Outro fator importante é que se trata de uma solução compatível com os ambientes *Linux*, *Macintosh* e *Windows*. Todas as interações com o Núcleo do sistema virtual utilizam a estrutura do laboratório de corrente alternada, figura 4.10.

O ACLab é a estrutura de uma plataforma de trabalho composta por um repositório de componentes virtuais associados a seus respectivos métodos (descritos por comportamentos matemáticos), e os módulos de montagem.

A Plataforma de Trabalho está dividida em quatro sessões principais:

1. um PBV (Protoboard Virtual);

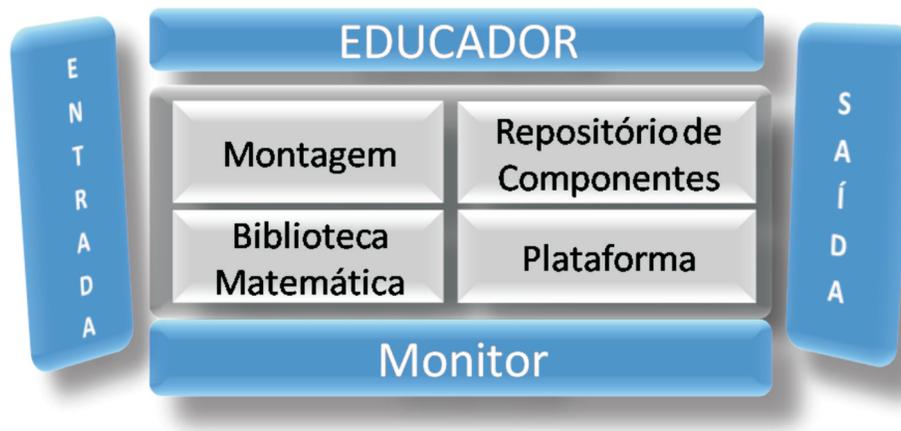


Fig. 4.10: Estrutura do Laboratório AC

2. um menu de elementos virtuais (Biblioteca de Signos);
3. uma sessão de telas e interfaces (Biblioteca de Medidores);
4. um painel de controle e configuração.

1. O PBV (Protoboard Virtual) é a matriz de células onde se pode montar os circuitos, utilizando componentes virtuais, figura 4.11. A versão implementada é um conjunto de 120 unidades PV12 interligadas em uma matriz de 8 linhas por 15 colunas. Associado a esta base de configurações existe uma rotina de varredura matemática que atualiza os valores na plataforma. As posições ocupadas são identificadas e a natureza do componente vai determinar se haverá algum tratamento secundário a ser executado. Em caso positivo o controle é transferido para o elemento naquela posição, de forma que os sinais e as variáveis associadas a eles sejam modificados segundo suas funções características. Este processo secundário, normalmente recursivo, pode ser irradiado interrompendo a varredura padrão do PBV, que será retomada no retorno desta rotina secundária. A complexidade destes processos são compatíveis com o nível de atividade do componente e sua importância dentro do ambiente. Uma fonte, por exemplo pode interferir com os níveis de tensão em todo o circuito dependendo de suas características e de como está conectada aos demais componentes. Estes são ditos componentes ativos que podem propagar alterações por toda a plataforma de testes.

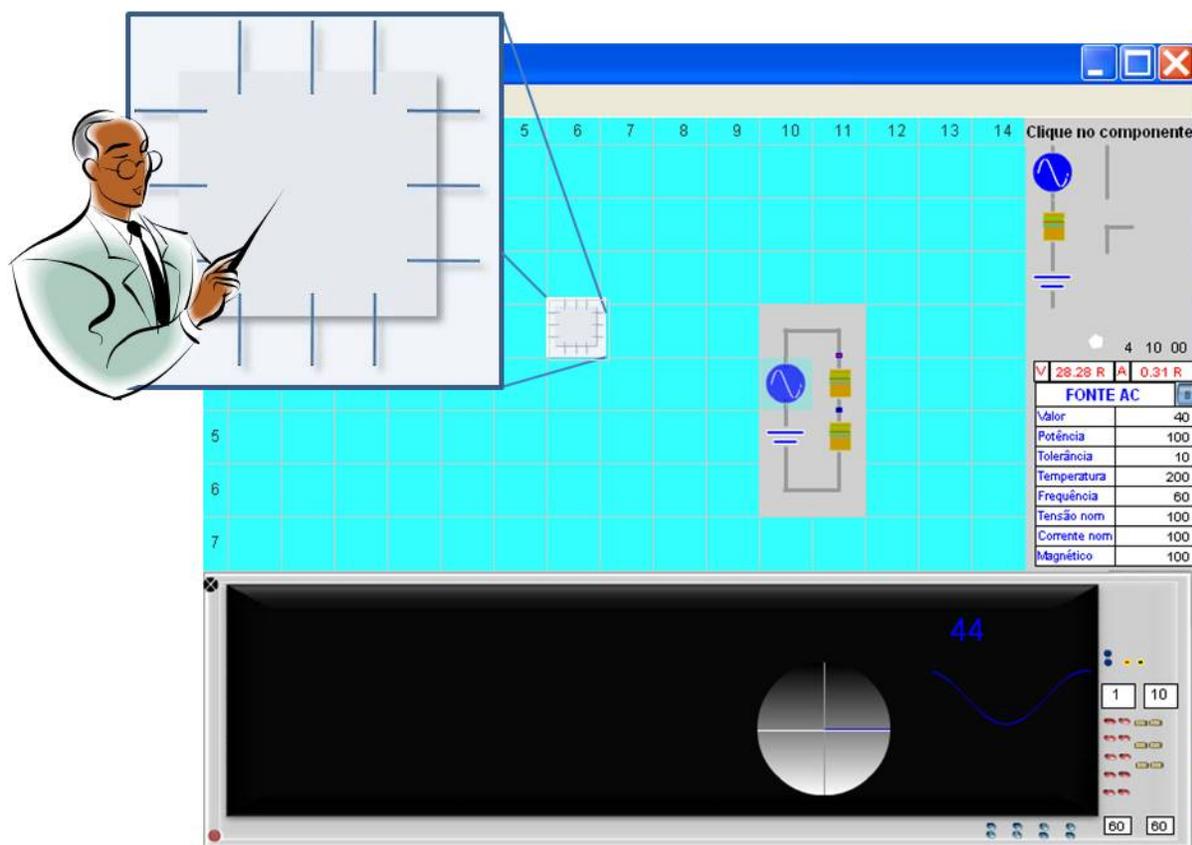


Fig. 4.11: Estrutura do Painel de Conexões Virtuais

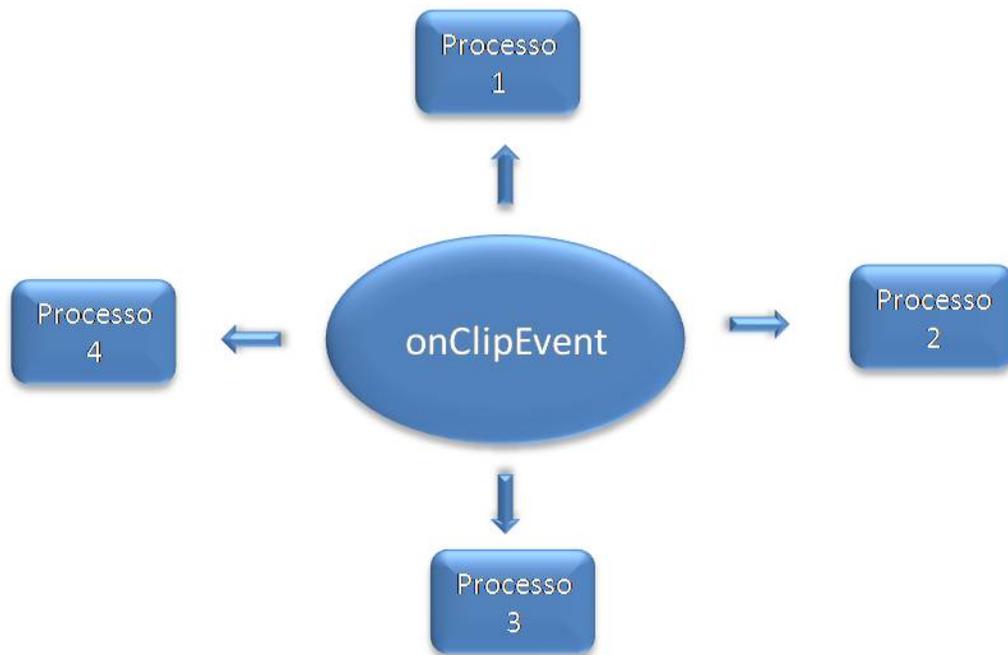


Fig. 4.12: Estrutura do Controlador onClipEvent, tipo Round-robin

O algoritmo 4, demonstra a execução recursiva da varredura matemática do Protoboard Virtual como um dos processos ativados pelo controlador tipo “Round-robin”, figura 4.12, que controla o sistema [Wik08].

2. Um menu de elementos virtuais do qual se pode duplicar e arrastar exemplares de Signos, por exemplo, figura 4.13;

A biblioteca de componentes virtuais é de fato muito mais, é o berço da criatividade e versatilidade de todo o sistema. Neste módulo reside a flexibilidade de códigos de programas, que podem ser carregados dinamicamente alterando o contexto e a finalidade do ambiente. As bibliotecas acomodam os componentes originais que poderão ser clonados (instanciados) a um toque do “mouse”. Estes componentes são associados a programas ditos “Descritivos de Signos” que, em sua estrutura, mantém as características do objeto, sua identidade visual (Ícone), sua topologia de conexões e sua rotina funcional que, com procedimentos matemáticos, sugerem como este signo deve interagir em um ambiente em que for inserido.

Permitir a criação de “Descritivos de Signos” é promover o desenvolvimento da criatividade, além de transmitir ao intérprete a responsabilidade pela semiótica [Not96]. A organização de um Descritivo de Signos pode ser vista na figura

Algoritmo 1: function calc(*cvpbl*,*cvpbc*,*s1*,*s4*,*s7*,*s10*)**Data:**

(*cvpbl*, *cvpbc*) = (1..8, 1..15) , posição (linha,coluna) no PBV

s1, *s4*, *s7*, *s10* = 0 ou 1 , pino de origem do sinal

Result:

circ = 0..2 , =0 aberto, =1 atualização, =2 modificado

begin

```

    circ ← 0 Inicia varredura sinalizando circuito ABERTO
    Captura a identificação do Signo, se houver
    fcomp ← root.vpb[cvpbl][cvpbc][root.vpbnter]
    Verifica se existe um componente nesta posição do PBV
if fcomp ≠ 0 and (cvpbl, cvpbc) ∈ PBV then
    Se voltar ao Signo inicial, sinaliza circuito fechado e sai
    if (cvpbl, cvpbc) = (root.bl, root.bc) then
        if root.circuit = 0 then
            ⊥ root.circuit ← 1
            Então, trata-se de um novo Signo no Protoboard Virtual
        else
            vpb[cvpbl][cvpbc][13] ← root.ativo+ = 1
            if s1 ≠ 0 then
                ⊥ vpb[cvpbl][cvpbc][8, 7, 6] ← vpb[cvpbl][cvpbc + 1][0, 1, 2]
            if s4 ≠ 0 then
                ⊥ vpb[cvpbl][cvpbc][11, 10, 9] ← vpb[cvpbl][cvpbc + 1][3, 4, 5]
            if s7 ≠ 0 then
                ⊥ vpb[cvpbl][cvpbc][2, 1, 0] ← vpb[cvpbl][cvpbc + 1][6, 7, 8]
            if s10 ≠ 0 then
                ⊥ vpb[cvpbl][cvpbc][5, 4, 3] ← vpb[cvpbl][cvpbc + 1][9, 10, 11]
            Executa-se a rotina matemática através do apontador “fcomp”
            fcomp.comp()
            if fcomp.s1 ≠ 0 then
                circ+ = calc(cvpbl, cvpbc - 1, 1, 0, 0, 0)
            else
                if fcomp.s4 ≠ 0 then
                    circ+ = calc(cvpbl + 1, cvpbc, 0, 1, 0, 0)
                else
                    if fcomp.s7 ≠ 0 then
                        circ+ = calc(cvpbl, cvpbc + 1, 0, 0, 1, 0)
                    else
                        if fcomp.s10 ≠ 0 then
                            ⊥ circ+ = calc(cvpbl - 1, cvpbc, 0, 0, 0, 1)
            if circ = 0 then
                ⊥ root.circuito = 0
    return(circ)

```

end

Algoritmo 2: function vercircuito()

```

begin
  atv ← vpb[root.bl][root.bc][root.vpbnter + 1] Nível de atividade do
  componente
  componente ← vpb[root.bl][root.bc][root.vpbnter] Identificação do
  componente
  root.ccomp ← 1 Inicia o contador de Signos
  root.ativo ← 1 Inicia o identificador de Signo ativo
  Verifica-se a rotação do Signo
  if componente.rotation = 0 and circuito = FECHADO then
    Se Signo na posição vertical, sinal vem de baixo
    componente.invpter ← 4
    vpb[root.bl][root.bc][4] ← vpb[root.bl + 1][root.bc][10]
    componente.comp()
    circ = calc(r.oot.bl - 1, r.oot.bc, 0, 0, 0, 1)
  else
    if componente.rotation = 90 and circuito = FECHADO then
      Se Horizontal a direita, sinal vem da esquerda
      componente.invpter ← 1
      vpb[root.bl][root.bc][1] = vpb[root.bl][root.bc - 1][7]
      componente.comp()
      circ = calc(r.oot.bl, r.oot.bc + 1, 0, 0, 1, 0)
    else
      if componente.rotation = 180 and circuito = FECHADO
      then
        Vertical para baixo, sinal vem de cima
        componente.invpter ← 10
        vpb[root.bl][root.bc][10] = vpb[root.bl - 1][root.bc][4]
        componente.comp()
        circ = calc(r.oot.bl + 1, r.oot.bc, 0, 1, 0, 0)
      else
        if componente.rotation = -90 and
        circuito = FECHADO then
          Horizontal a esquerda, sinal vem da direita
          componente.invpter ← 7
          vpb[root.bl][root.bc][7] = vpb[root.bl][root.bc + 1][1]
          componente.comp()
          circ = calc(r.oot.bl, r.oot.bc - 1, 1, 0, 0, 0)
        end
      end
    end
  end
end

```

Algoritmo 3: function Matscroll()

```

begin
  Matini ← False
  Faz-se a varredura de todos os componentes ativos pela
  grandeza 1, mais ativo (fontes), 99 menos ativos
  for root.bc = 0; root.bc < root.vpbncol; root.bc ++ do
    for root.bl = 0; root.bl < root.vpbnlin; root.bl ++ do
      Identifica se é uma fonte, atv=1
      if vpb[root.bl][root.bc][root.vpbnter + 1] = 1 and Matini = False
      then
        Matini ← True
        Sinaliza circuito aberto
        root.circuito ← 0
        root.resistencia ← 0
        root.voltagem ← 0
        root.voltAC ← 0
        root.corrente ← 0
        Incrementa-se o tempo na frequencia de 60Hz
        root.systempo+ = 60/20
        Verifica-se o circuito, se existir
        vercircuito()
        if root.circuito = 1 then
          Circuito foi fechado, precisa ser repolarizado
          vercircuito()
          root.corrente ←
            (root.voltagem + root.voltAC)/root.resistencia
          Sinaliza circuito repolarizado
          root.circuito ← 2
          Propaga o sinal da fonte principal
          vercircuito()
    end
  end
end

```

Algoritmo 4: *onClipEvent(enterframe)*—Processo principal do Protoboard Virtual

```

begin
  Guarda-se o estado de circuito fechado ou aberto
  testcirc ← root.circuito
  Chama-se a rotina de varredura matemática
  root.Matscroll()
  Verifica se existe um circuito mínimo
  if root.circuito ≥ 2 and root.ccomp > 5 then
    if root.resistencia ≠ 0 then
      root.corrente ← (root.voltagem + root.voltAC)/root.resistencia
      root.icorren ← root.corrente
    else
      root.icorren ← CURTO
    else
      root.circuito ← 0
      root.icorren ← ABERTO
      Verifica se o circuito foi aberto após um fechamento
      if testcirc ≠ 0 then
        Limpa-se os potenciais residuais do VPB
        root.LimpaSinaisVPB()
      root.atualiza ← False
  end

```

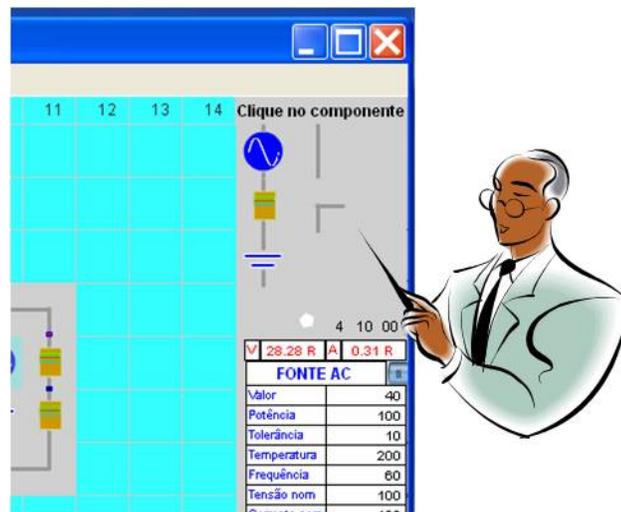


Fig. 4.13: Repositório de Signos

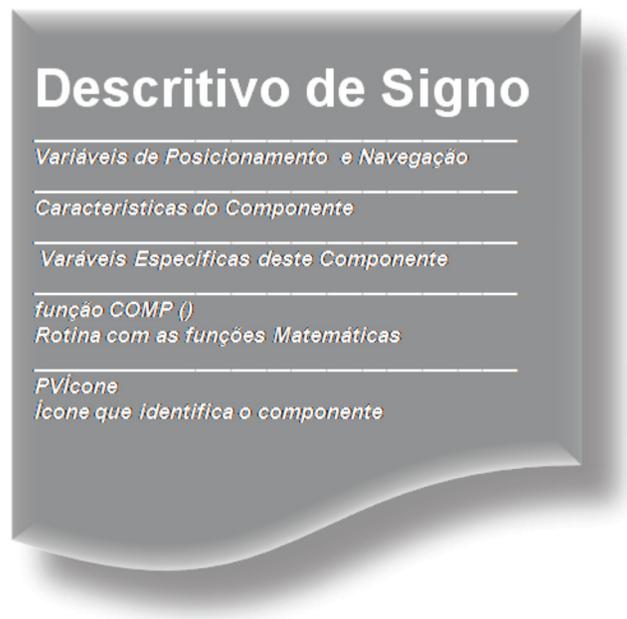


Fig. 4.14: Estrutura do Signo

4.14 e é composta de:

- (a) Conteúdo de navegação, que armazena os indicadores de posicionamento e do fluxo de informações no componente;
- (b) Características, nível onde relaciona as informações relativas ao objeto e que serão visíveis ao ambiente como Corrente Máxima, Temperatura Máxima e outras;
- (c) Variáveis Específicas, onde se controla parâmetros específicos deste tipo de elemento, se houver, como tensão eficaz (RMS) por exemplo no caso de uma componente de Corrente Alternada;
- (d) Função “comp()”, rotina acionada por apontadores com alocação dinâmica que é replicada no conceito de geração e manipulação de objetos. Permite o desenvolvimento de comportamentos diversos através de trechos de código que executam funções matemáticas. Estes programas permitem que se emule um resistor, um sistema massa-mola, uma reação química e até mesmo um Memristor, [HPL08], ou outro componente ainda imaginário;
- (e) PVÍcone, instrumento de identificação visual utilizado para facilitar a interação com o ambiente, o sistema de monitoração e outros componentes. Pode ser implementado em três dimensões dando acesso a camadas de

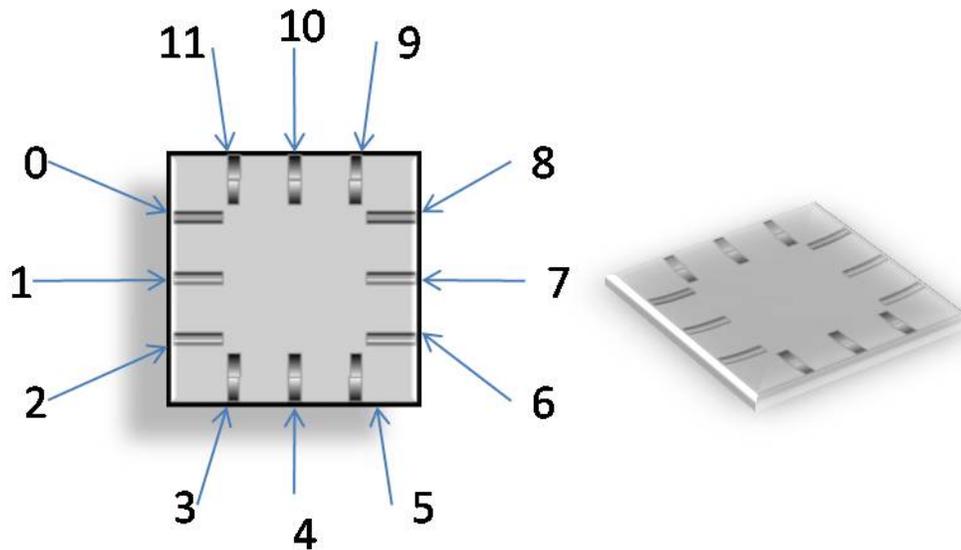


Fig. 4.15: PVÍcone

informação e de interação, figura 4.15.

A utilização do PVÍcone para representação visual dos signos pode utilizar o recurso de multicamadas. Com este artefato pode-se agregar ao objeto outros níveis funcionais além do diagrama de conexões, é possível implementar um nível de controle do signo, por exemplo, onde se pode movê-lo, rotacioná-lo e até mesmo excluí-lo, figura 4.16. Neste exemplo também foi incluído um campo de identificação do componente opcional, no mesmo nível. Este é um exemplo da potencialidade deste recurso. A característica de várias dimensões permite que se implementem funções em outras camadas, representando assim, comportamentos paralelos e interações subliminares ao cenário que se está visualizando.

3. Uma sessão de telas e interfaces, que fazem parte do sistema de monitoramento para mostrar o comportamento dos sinais, figura 4.17. A formação do conhecimento normalmente ocorre através da coleta e da associação de informações. Ao se identificar relações que modificam os signos, pode-se dar então um significado para algum fenômeno observado. Portanto, mais do que desenvolver um ambiente virtual para simulação e testes, é necessário que se tenha aparelhos didáticos de observação deste ambiente. O rendimento do processo cognitivo estará diretamente ligado à eficiência destes aparelhos em demonstrar os fenômenos, deixando as conclusões para serem formadas na mente do intérprete. A simulação de aparelhos conhecidos pode facilitar a visualização e

Algoritmo 5: *onClipEvent(load)* – Instância do Signo RESISTOR

```

begin
  Nome do Signo e posição dele na VPB
  vnome ← "RESISTOR"
  inicia a posição do Signo na linha 1 e na coluna 15
  vlinha ← 1
  vcolum ← 15
  Inicia variáveis específicas deste Signo
  icorren ← 0 - Corrente
  ivoltag ← 0 - Voltagem
  flagshow ← 0 - Visibilidade
  irms ← 0 - Corrente efetiva
  vrms ← 0 - Tensão efetiva
  Inicia variáveis de Navegação do Signo
  invpter ← 4 - Terminal de entrada Default
  Inicia variáveis de Navegação do Signo
  s1 ← 0 - Sentido de propagação a esquerda
  s4 ← 0 - Sentido de propagação para baixo
  s7 ← 0 - Sentido de propagação a direita
  s10 ← 0 - Sentido de propagação para cima
  Inicia os "prompts" do painel de contrôle
  carac0 ← "Valor"
  carac1 ← "Potncia"
  carac2 ← "Tolerncia"
  carac3 ← "Temperatura"
  carac4 ← "Freq.mx"
  carac5 ← "Tensomx"
  carac6 ← "Correntemx"
  carac7 ← "Camp.Magnt."
  carac8 ← "Linha"
  carac9 ← "Coluna"
  Inicia os valores no Painel de Contrôle
  vvalor ← 100 - Valor do Resistor em ohms
  vpoten ← 200 - Potência Máxima em Watts
  vtoler ← 0.10 - Tolerância, 10 por cento
  vtempe ← 90 - Temperatura Máxima
  vfrequ ← 1000 - Frequência Máxima
  vtensa ← 100 Tensão Máxima
  vcorre ← 2 Corrente Máxima
  vmagne ← 2000 Limite de Campo Magnético
  // Controle de Visibilidade, quando seleciona-se com o mouse
  this.dadcomp.visible ← false
end

```

Algoritmo 6: *fcomp()* – Modelamento Matemático do Signo

```

begin
  Verifica se é uma instância do Signo ou é o Original
  if this.name.substring(0,6) = "cpiece" then
    Identifica o sentido de propagação no componente
    this.rotation pode ser (0, 90, 180, -90)
    sentido ←  $-3 * (this.rotation/90)$ 
    vpbter10 ←  $10 + sentido$  vpbter10 é o terminal de saída
    sentido pode ser -3, -6, 3, 0
    vpbter4 ←  $4 + sentido$  vpbter4 é o terminal de entrada
    if vpbter4 < 0 then vpbter4+ = 12
    if vpbter10 > 11 then vpbter10- = 12
    Limpa indicadores e informa o sentido de propagação do sinal
    this.s1 ← this.s4 ← this.s7 ← this.s10 ← 0
    if this.invpter = 1 then this.s7 ← 1 a esquerda
    if this.invpter = 4 then this.s10 ← 1 para baixo
    if this.invpter = 7 then this.s1 ← 1 a direita
    if this.invpter = 10 then this.s4 ← 1 para cima
    Processa as informações
    if this.invpter = vpbter4 then
      vpb[vlin][vcol][vpbter10] = vpb[vlin][vcol][vpbter4] - ivoltag
    else
      vpb[vlin][vcol][vpbter4] = vpb[vlin][vcol][vpbter10] - ivoltag
    if this.vvalor ≠ 0 then
      icorren ← ivoltag/this.vvalor
    if root.voltAC ≠ 0 then
      if root.atualiza = True then
        this.vrms ← this.irms ← 0
        newvrms ← ivoltag
        if newvrms > this.vrms then
          this.vrms ← newvrms
          v ←  $ivoltag / \sqrt{2} + "R"$ 
        else v ←  $this.vrms / \sqrt{2} + "R"$ 
        newirms ← root.corrente
        if newirms > this.irms then
          this.irms ← newirms
          i ←  $root.corrente / \sqrt{2} + "R"$ 
        else i ←  $this.irms / \sqrt{2} + "R"$ 
        else
          v ← ivoltag and i ← root.corrente
      if this.flagshow = 1 then root.ivoltag ← v and root.icorren ← i
      Acrescenta a resistência total do circuito
      if root.circuito = 1 then root.resistencia+ = this.vvalor
end

```

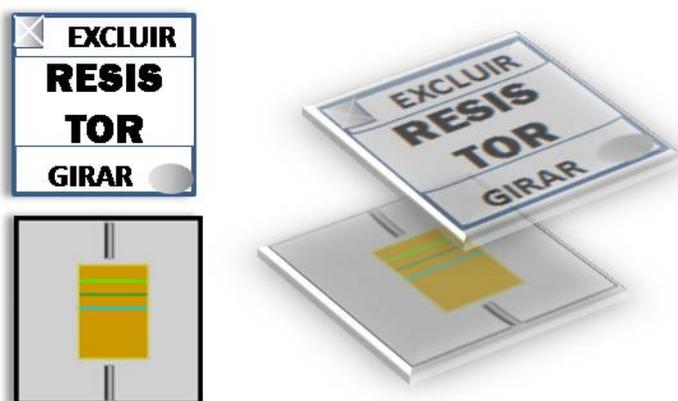


Fig. 4.16: Resistor utilizando um PVÍcone com duas camadas, pinos 4 e 10

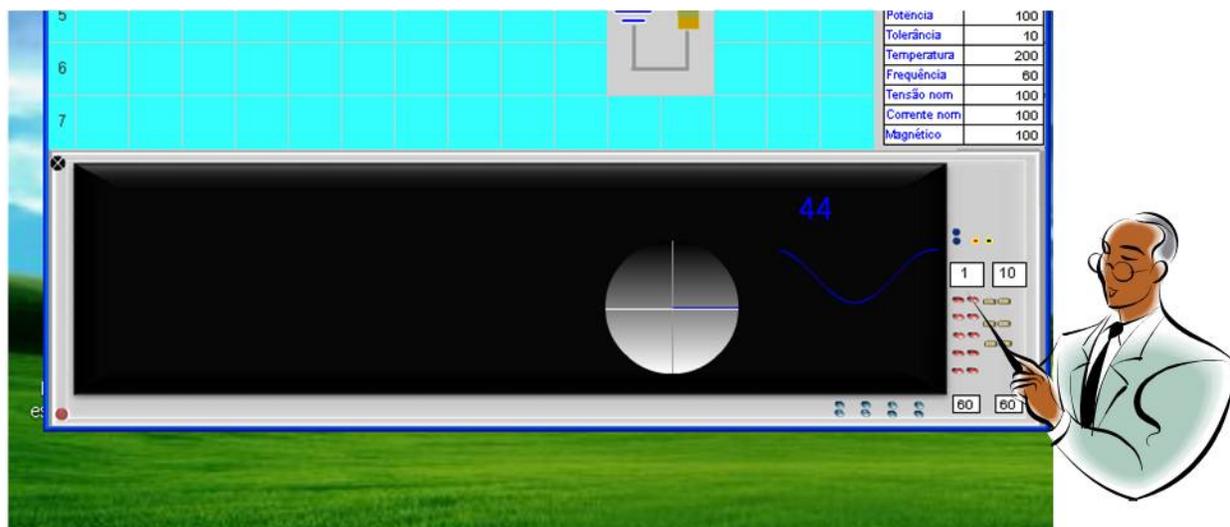


Fig. 4.17: Dispositivos de Monitoração

a medição de algumas características ou alterações no ambiente que se deseja observar. Estudar os fenômenos pode ajudar a compreender a natureza dos componentes e sua interação com os demais objetos da biblioteca virtual. Mas também pode-se criar equipamentos ainda não desenvolvidos para demonstrar algumas características igualmente abstratas como, por exemplo, as ondas eletro-magnéticas, e os Memristores [HPL08], ampliando assim os horizontes da criação e do conhecimento. Dentro de uma visão pós-modernista, é possível desafiar os paradigmas existentes, propor novas abordagens para temas conhecidos e até mesmo para outros ainda pouco explorados. Equipamentos especiais podem também promover análises comparativas entre sinais simulados difíceis de serem implementados na prática.

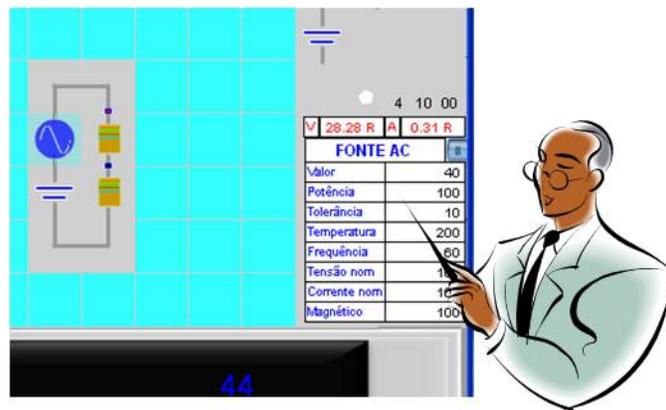


Fig. 4.18: Painel de Interação

4. Um painel de controle e configuração, com comandos para gerenciar o ambiente de testes, modificando algumas características dos componentes virtuais montados no PBV, figura 4.18. A interação com os objetos permite ao intérprete, conforme proposto em [Not95], a validação de novos signos. Um signo pode ser formado de maneira diferente na mente de um interpretante e de outro. Esta capacidade de mutação dos signos na mente de um interpretante está, normalmente, associada à maturidade do processo cognitivo. A interação do interpretante com o ambiente pode facilitar esta análise e a convergência do entendimento. Um painel de configurações que permita alterar as características dos objetos no ambiente virtual, vai possibilitar a validação ou negação de hipóteses acelerando o raciocínio lógico dedutivo. Este painel deve permitir que se consulte estados dos objetos e eventualmente que se possa acompanhar algum fenômeno específico como temperatura, vibração, campo magnético, corrente elétrica e outros. As alterações podem ser verificadas em função da resposta do objeto a variações submetidas através do painel de controle e permitem ao aluno desenvolver uma teoria a respeito do objeto ou do circuito em estudo.

A biblioteca de modelos matemáticos que, associada ao repositório de componentes, será usada para definir as características deste elemento. O módulo de montagem é responsável por todos os procedimentos de configuração e teste dos esquemas virtuais. Ele é também responsável pelo processo de varredura matemática, descrito anteriormente, de modo a simular o comportamento do circuito a ser estudado.

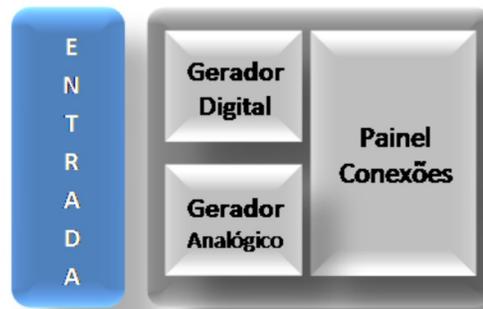


Fig. 4.19: Estrutura do Módulo de Entrada

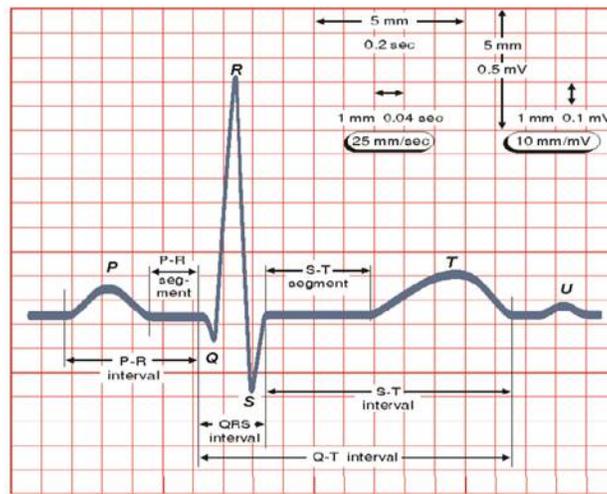


Fig. 4.20: Onda de ECG como sinal de Entrada

4.3.4 Módulo de Entrada

O circuito em análise pode receber sinais externos que, podem ainda variar segundo outros parâmetros não controlados pelo circuito em estudo. Normalmente a fonte de sinal será interna e portanto gerada a partir de componentes virtuais presentes na biblioteca de componentes, figura 4.13. Em um ambiente de teste deve-se poder gerar vários sinais de alimentação com diferentes formatos, figura 4.19. Gerar e conectar sinais aos circuitos são as tarefas principais do módulo de entrada.

Em geral, utiliza-se formas de onda tradicionais nos sinais estudados, como senóides, sinais contínuos, ondas quadradas ou dente de serra, mas algumas vezes precisa-se utilizar alguns sinais pré-processados, mais complexos do que o normal, como por exemplo uma onda de ECG (Eletrocardiograma), figura 4.20. Adicionalmente, pode-se prever um painel de conexões para interligar diferentes tipos de equipamentos ao ambiente de testes.



Fig. 4.21: Módulo Educador

4.3.5 Módulo Educador

Uma biblioteca de lições e conceitos pré-definidos pode apoiar os estudantes em alguns testes e experiências. Os processos de apoio a lições pré-selecionadas são conduzidos pelo módulo Educador.

Máscaras com sugestões de configuração de circuitos podem ser apresentadas no cenário e orientar montagens específicas. Este tipo de orientação pretende demonstrar alguma hipótese que leve à dedução de algum conhecimento que se deseja transmitir. Além disso o gerenciador de configurações irá atuar como suporte do educador durante o processo de montagem, figura 4.21. Utilizando uma base de dados, o módulo de consultoria pode ajudar o estudante enviando mensagens de alerta e outros eventos animados durante os experimentos, enriquecendo desta forma os cenários educacionais.

4.3.6 Módulo de Monitoramento

Uma grande contribuição que se pode obter dos sistemas elétricos virtuais é a capacidade de se poder observar os valores e o comportamento dos sinais, figura 4.22. Desta forma pode-se demonstrar cenários e conclusões referentes a algum efeito especial específico. Formas de onda são verificadas utilizando a estrutura de monitoração. A seção de *SETUP* irá definir que tipos de equipamentos e que configurações serão utilizadas para monitorar o circuito. Os analisadores e medidores são conectados a pontos do circuito virtual para capturar informações dos sinais que são submetidos a sessões de apresentação de acordo com ajustes efetuados nos equipamentos medidores.



Fig. 4.22: Estrutura do módulo de monitoração

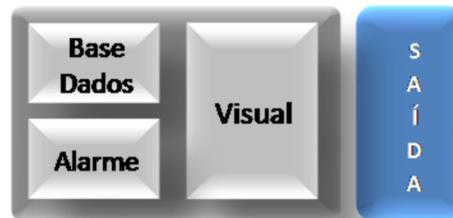


Fig. 4.23: Dispositivos de Saída

4.3.7 Módulo de Saída

Quando se pretende simular circuitos e situações de teste, deseja-se as vezes obter os resultados através de algum dispositivo de saída, como alarmes, sensores, atuadores e outros. Os resultados podem ainda ser coletados e armazenados através de dispositivos de saída, figura 4.23. Além do ambiente virtual pode-se submeter os efeitos e as respostas do circuito a controladores e atuadores ou acender, até mesmo a distância, LEDs em dispositivos de saída.

4.4 Criando um objeto sígnico

O objeto que se pretende incorporar a uma biblioteca de signos deve obedecer a estrutura de formação deste conjunto de elementos. A estrutura de formação é definida pela orientação pedagógica que se deseja propor através do objeto de aprendizagem. Neste caso serão utilizados signos compostos por elementos com duas camadas. A primeira incorpora as funções de informação, movimentação, rotação e exclusão. A segunda camada representa a topologia do signo e suas características de propagação da informação, no ambiente em que for inserido. Estas funções estão associadas ao PVÍcone (Ícone do Painel Virtual) que, interagindo com o mouse ou

com outros *artefatos pedagógicos*, permitem ao aluno colher informações e explorar seu comportamento em um determinado ambiente, cooperando ou não com outros signos colocados no PBV. A seqüência de criação é descrita no algoritmo 7, a seguir.

Algoritmo 7: Passos para Criação de um Signo

beginUtilizando-se um gerador de programas FLASH cria-se o código em *Action Script*

1 Utiliza-se um Painel Virtual original para definir os contatos de interação do Signo

2 Cria-se o Ícone que irá representar o Signo

3 Codifica-se o algoritmo 5, que inicia o signo e suas instâncias

4 Codifica-se o algoritmo 6, que é a função de comportamento do Signo

end

4.5 Estudo de Casos

Para que se permita ilustrar a flexibilidade destas soluções serão apresentados alguns estudos de casos. Estes ensaios virtuais tem objetivo pedagógico e podem ainda ser demonstrados utilizando-se lousas eletrônicas como ferramentas de apoio ao ensino presencial. Para efeito do tema eletrotécnica serão utilizados signos que representam componentes e configurações básicas desta área que, no entanto, utilizam as estruturas conceituais e os recursos propostos nos capítulos anteriores.

1. Circuito Aberto
2. Fonte Contínua e Resistor
3. Fonte Alternada com nível DC e Resistores em Série
4. Fonte Alternada e Resistores em Paralelo
5. Fonte Alternada e Capacitores

4.5.1 Circuito Aberto

Em um ambiente de simulação utilizado como objeto de aprendizagem, existem detalhes importantes mesmo na montagem dos cenários. Um exemplo é o comportamento de um circuito aberto, que ao contrário do que se imagina pode ser mais

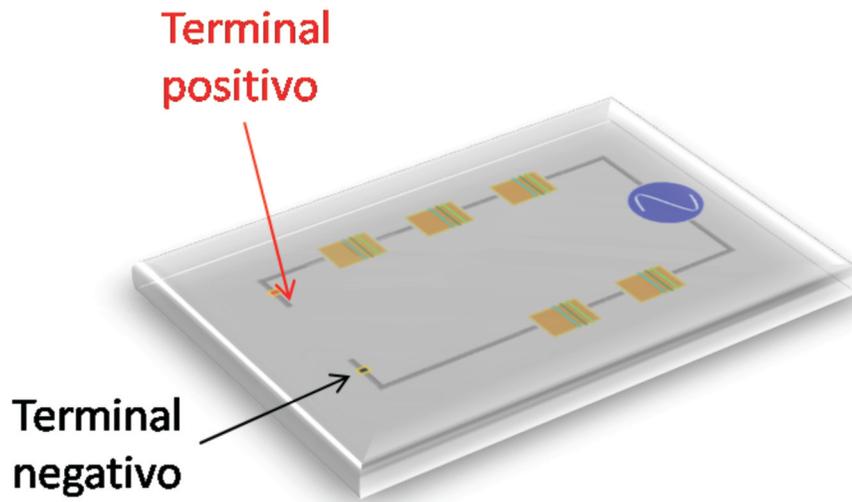


Fig. 4.24: Circuito Aberto

complexo de se polarizar do que um circuito fechado. De fato ao se analisar o circuito com uma rotina de varredura bidimensional não se sabe antecipadamente se o circuito está ou não fechado em alguma posição adiante. Esta questão força a utilização de uma rotina recursiva de exploração do cenário até que se consiga a resposta a esta questão, quando então todo o circuito precisa ser novamente analisado e polarizado. A presença de dispositivos considerados ativos, como uma fonte, também é imprescindível para se efetuar o cálculo de distribuição de tensão no circuito simulado, figura 4.24. Para a montagem do circuito aberto foram instanciados a fonte alternada de 40 Volts, cinco Resistores de 100 Ohms e os elementos de conexão necessários para a demonstração.

Através dos terminais positivo (Vermelho) e negativo (Preto) do canal Amarelo verifica-se a ausência da queda de tensão nas várias posições de um circuito aberto.

4.5.2 Fonte Contínua e Resistor

Neste ensaio utiliza-se um circuito fechado simples, para demonstrar a queda de tensão sobre o resistor, figura 4.25. Fez-se uso também de outros recursos como o módulo de monitoração que tem aqui implementado um analisador de sinais de dois canais (Azul e Amarelo). Neste caso utilizou-se somente o canal Azul do analisador para medição, figura 4.26. Para a montagem do circuito virtual foram instanciados a fonte contínua de 48 Volts, um Resistor de 100 Ohms e os elementos de conexão

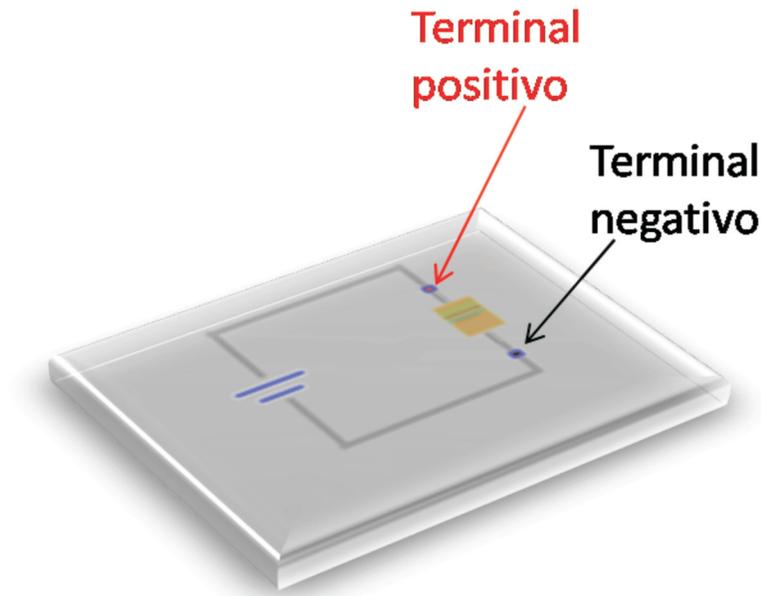


Fig. 4.25: Fonte Contínua e Resistor

necessários. Suas cópias foram arrastadas e arranjadas no PBV de forma a permitir a simulação de circuito que interliga a fonte e o resistor. Os resultados podem ser vistos no campo numérico e no formato de onda, ambos representados na tela do analisador quando ligado e tendo seus terminais de medição devidamente posicionados no circuito. Além disso demonstra-se o acesso às especificações de um componente, neste caso a fonte, quando o mesmo é selecionado com o “*mouse*”, figura 4.26, inclusive com a possibilidade de se alterar os valores de especificação da fonte para que os possíveis reflexos, no restante do circuito, possam ser monitorados.

Como integrante da biblioteca de equipamentos do módulo de monitoração, este analisador de sinais pode demonstrar alguns dos recursos implementáveis como o selecionador de escalas em amplitude e a pausa para análise do sinal. Através dos terminais positivo (Vermelho) e negativo (Preto) de cada canal (Azul ou Amarelo) pode-se simular o contato elétrico destes com as diversas partes do circuito e verificar-se a diferença de potencial entre pontos de conexão destes terminais. A interpretação que cada pessoa dá aos elementos da biblioteca de componentes é que fundamenta o conceito de “Signo” proposto por Peirce, [Not95], e que evolui em uma espiral onde o aluno (intérprete) adquire o conhecimento na medida em que descobre o significado daquele símbolo.

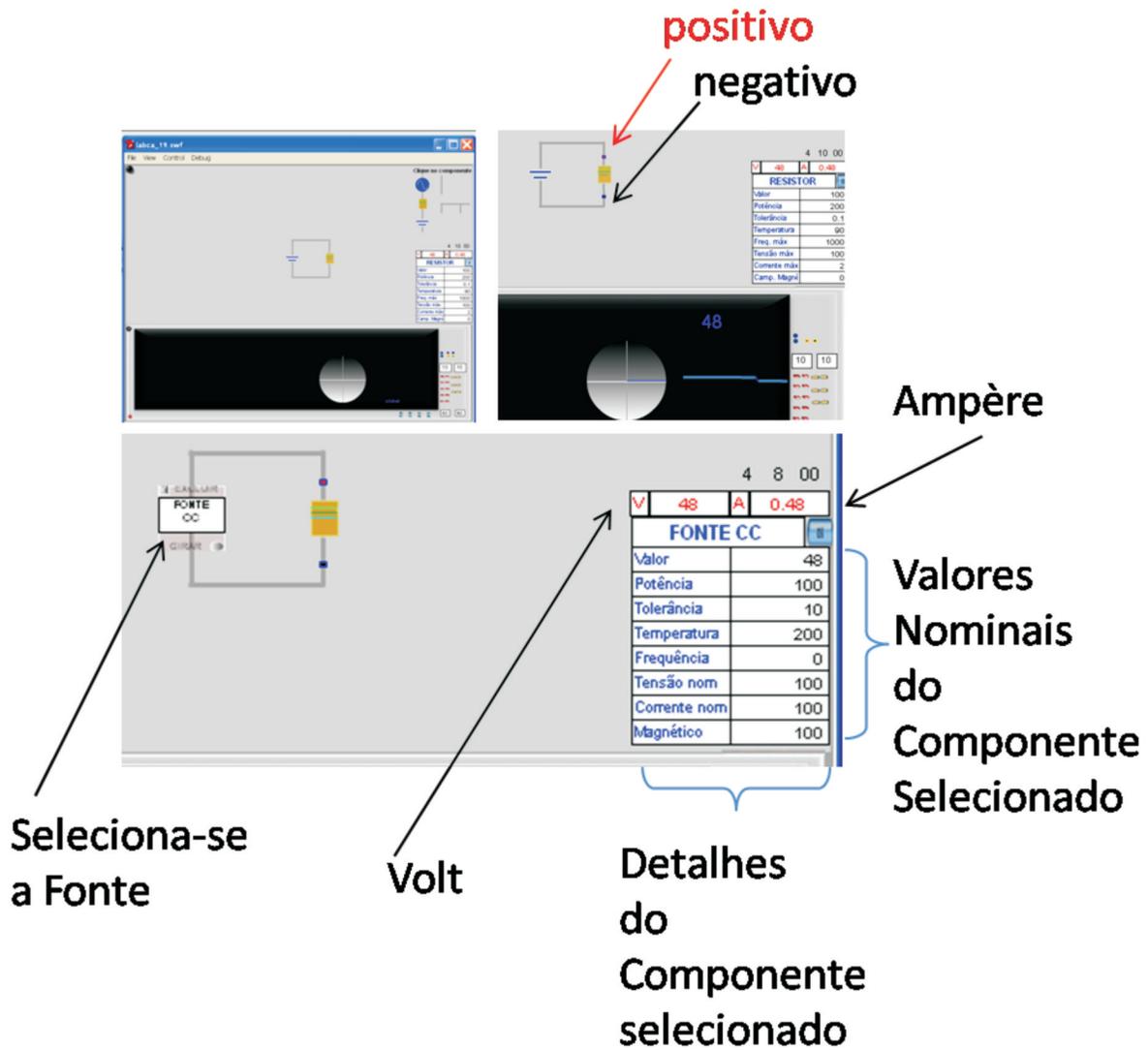


Fig. 4.26: Monitoramento de caso

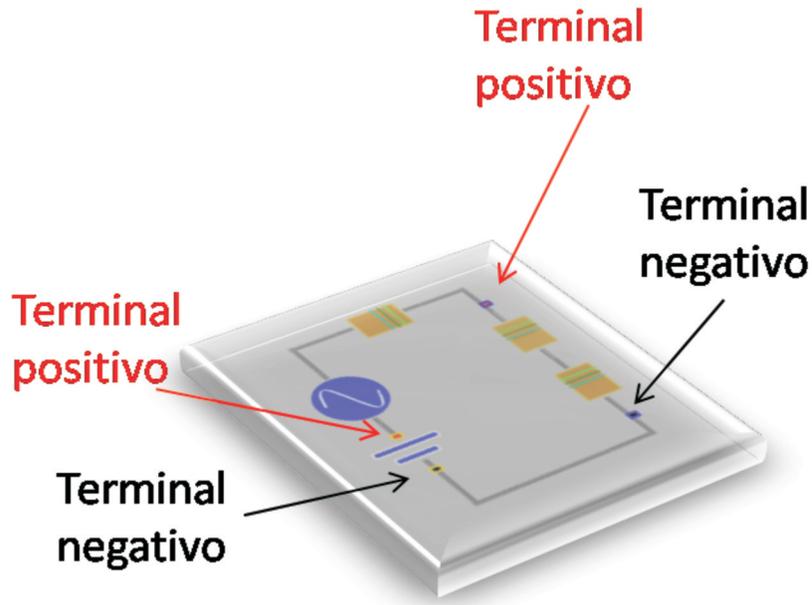


Fig. 4.27: Fonte AC com nível DC e Resistores em Série

4.5.3 Fonte AC com nível DC e Resistores em Série

Neste outro ensaio utiliza-se um circuito fechado com duas fontes e três resistores em série, para se demonstrar a composição de uma fonte de sinal senoidal e outra de sinal constante, bem como os efeitos destas na diferença de potencial sobre o resistores, figura 4.27. Neste caso o sistema automaticamente passa a representar no painel de controle os sinais eficazes (RMS) já que flutuam devido à componente senoidal, figura 4.29. Aqui utilizou-se os dois canais Azul e Amarelo do analisador para medição, sendo que os terminais de medição amarelos monitoram a tensão nos pólos da fonte de corrente contínua e os terminais azuis verificam a variação de tensão sobre dois dos três resistores. Observa-se que o canal amarelo está em escala de amplitude 1/10 e o azul em 1/1. Para a montagem do circuito virtual foram instanciados a fonte contínua de 48 Volts, a fonte senoidal com 40 Volts de pico, três resistores de 100 Ohms e os elementos de conexão necessários. Suas cópias foram arrastadas e arranjadas no PBV de forma a permitir a simulação de circuito que interliga estes signos. Os resultados podem ser vistos no campo numérico e no formato de onda, ambos representados na tela do analisador, quando ligado e tendo seus terminais de medição devidamente posicionados no circuito.

O PBV dispõe de uma grade de orientação que pode ser habilitada ou não através do botão localizado no canto superior esquerdo do mesmo. Ao se movimentar o

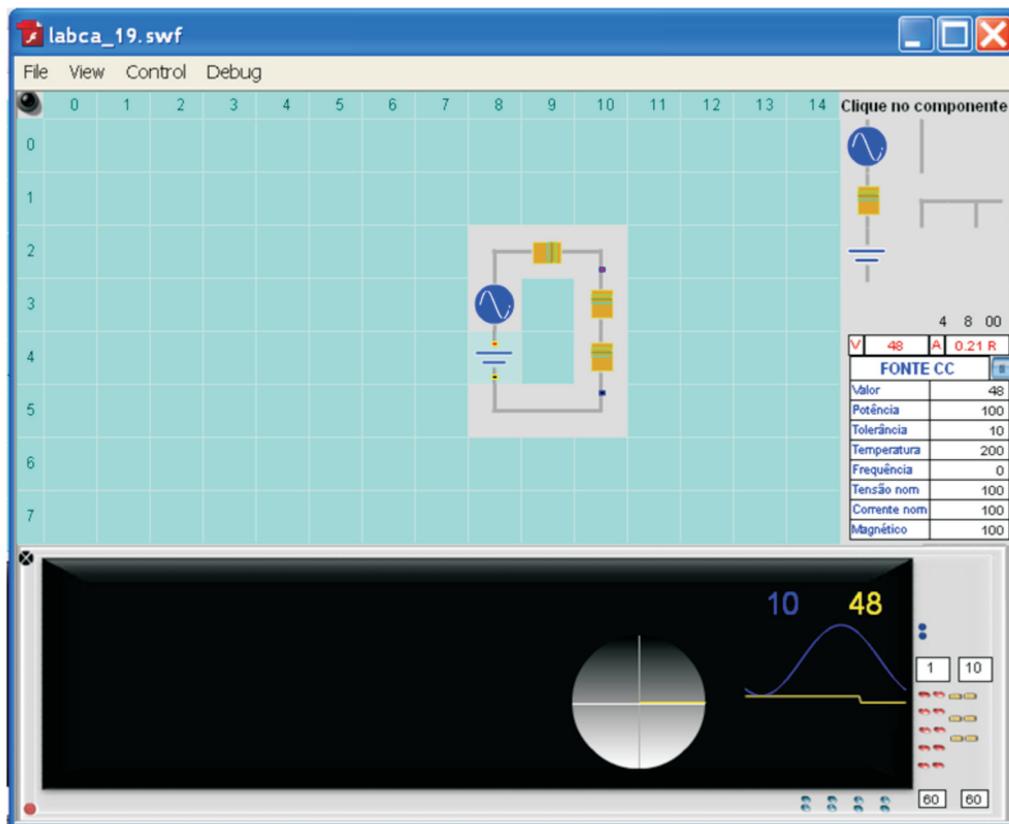


Fig. 4.28: Monitoramento de Caso

“mouse” sobre os componentes no PBV aciona-se a camada superior destes signos onde se verifica informações e os controles de rotação e exclusão destes. Também pode-se manter o botão esquerdo do mouse pressionado sobre o signo e assim deslocá-lo, se desejar.

Neste caso demonstra-se o acesso às especificações de um dos resistores quando o mesmo é selecionado com o “mouse”, figura 4.29, inclusive com a possibilidade de se alterar os valores de especificação do resistor para que os possíveis reflexos, no restante do circuito, possam ser monitorados. O procedimento de alteração das características dos componentes utiliza os campos numéricos do painel de controle e o botão azul de atualização no canto superior direito deste painel, que ao ser acionado pode atualizar os valores e alterar o comportamento do componente selecionado no PBV.

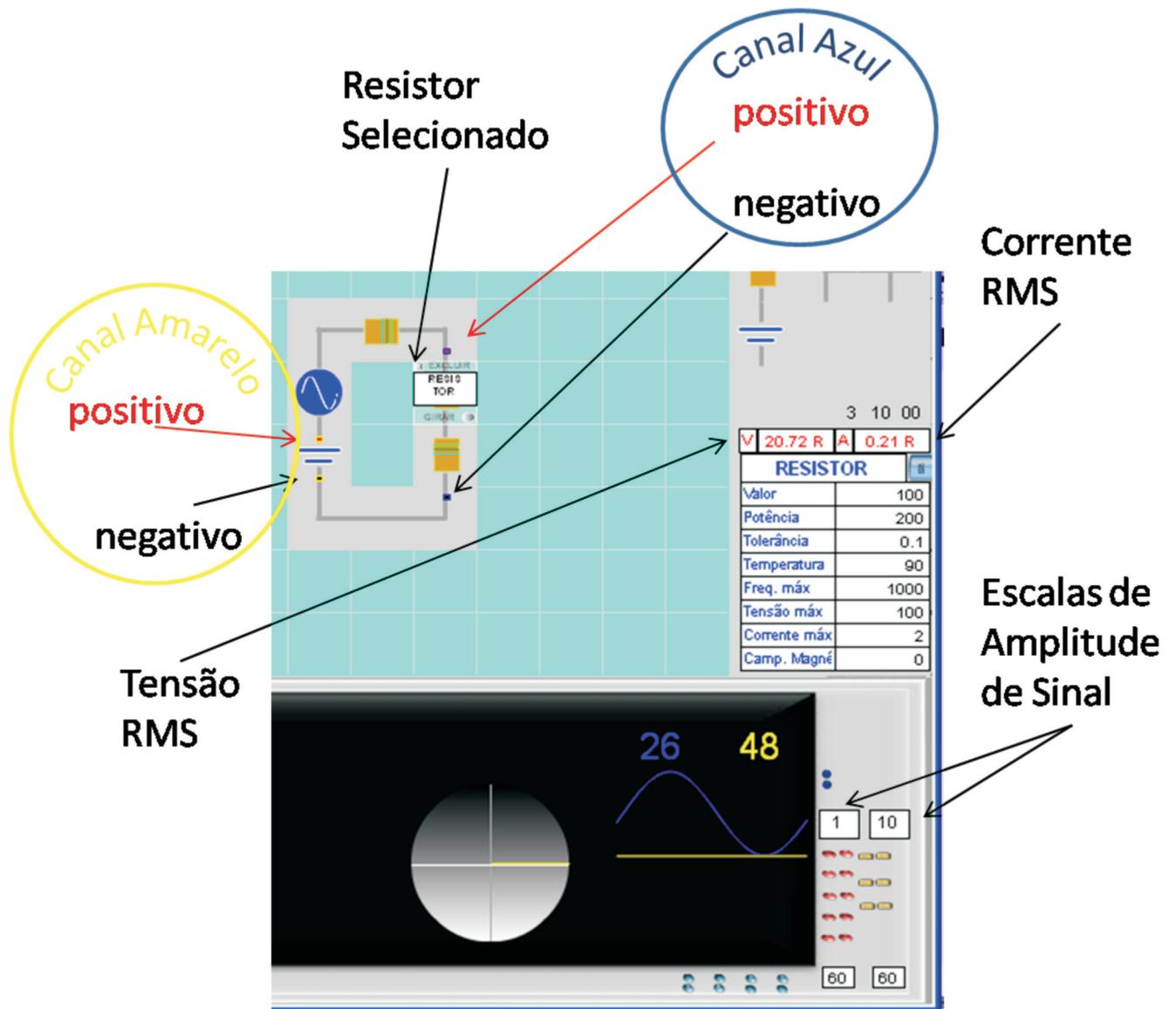


Fig. 4.29: Detalhes do Painel de Monitoração

4.6 Conclusão deste Capítulo

Neste capítulo apresentou-se o modelo tridimensional para o desenvolvimento de softwares educacionais com destaque para a implementação do núcleo do sistema. Foram descritas as funcionalidades básicas do mesmo e a potencialidade da biblioteca de signos, conforme proposto pela Semiótica de Peirce. Relevou-se também a modularidade, a portabilidade e a flexibilidade proporcionada pelo uso do “Flash” como ferramenta de desenvolvimento e das imagens vetoriais como adequadas para este tipo de aplicação. O módulo descritivo de signo, permite que lições educativas possam ser feitas em outras áreas como a química ou outras engenharias, de forma padronizada e até com pouco conhecimento de programação. O recurso de sub-níveis do PBV (Protoboard Virtual) e dos PVs (Painéis Virtuais) acrescentam uma característica de multi-identidade visual e interatividade para estes elementos. Objetos da biblioteca de signos foram instanciados, posicionados, rotacionados e deletados a um toque do mouse ou de apontadores, quando utilizados em conjunto com lousas eletrônicas especiais. Ao se implementar um analisador de sinais como exemplar da biblioteca de equipamentos virtuais, utilizou-se o sistema de monitoramento para complementar as lições demonstrando sinais elétricos e outros fenômenos em circuitos virtuais montados no PBV. Demonstrou-se inclusive, a versatilidade desta solução quando utilizada em sala de aula como ferramenta de apoio em conjunto com as lousas eletrônicas. No próximo capítulo, descreve-se a importância dos testes de Usabilidade principalmente em aplicações educacionais. Desenvolve-se então um modelo para testes de Usabilidade de softwares para Educação a Distância, o AVALUEAD. Em seguida utiliza-se este método para avaliação do Núcleo desenvolvido no capítulo anterior e apresenta-se os resultados.

Capítulo 5

USABILIDADE -

DESENVOLVENDO O AVALUEAD

Desenvolver programas de computador considerando requisitos de usabilidade, pode ser um fator decisivo para se alcançar alguns objetivos muito importantes de um projeto. Quando se fala em desenvolvimento de *software* para um ambiente educacional e possivelmente via Internet, é preciso garantir que estes sejam testados segundo critérios de avaliação de *software* WEB, com modelos de engenharia de *software* [Pre02] e critérios de usabilidade na WEB [Nie98, Nie02b], levando-se em consideração principalmente características específicas de um ambiente voltado à educação.

Usabilidade é um tema desenvolvido a partir dos estudos realizados por Dominique Scapin em 1990, visando a organização dos conhecimentos sobre ergonomia de interfaces homem-computador, onde foram desenvolvidos critérios de modo a facilitar o acesso e a utilização destes recursos. Através de experimentos variados, esse conjunto de critérios tem sido validado e aprimorado. A lista original de critérios definida em 1993 por Dominique Scapin e Christian Bastien apresentou um total de oito critérios principais, sendo que alguns deles se dividem em sub-critérios, que por sua vez se dividem em critérios elementares. Os critérios principais são os seguintes:

- Condução,
- Carga de Trabalho,

- Controle Explícito,
- Adaptabilidade,
- Gestão de Erros,
- Consistência,
- Significado dos Códigos e
- Compatibilidade.

A utilização de ferramentas versáteis como “*FLASH*” leva ainda a considerações adicionais e específicas [Mar05] que permitem extrair melhores resultados, podendo até facilitar a inclusão de alunos com necessidades especiais que precisam de recursos adicionais [Coy08] para interagirem com estes sistemas.

Vários métodos e técnicas podem ser utilizadas para elaboração de testes de validação de uma ferramenta de *software*. A seguir são apresentadas algumas técnicas conhecidas e detalhados somente os resultados obtidos pelo método de avaliação escolhido.

ErgoList

O projeto *ErgoList* [UFS98] tem por objetivo conceber, projetar, desenvolver e disponibilizar via Web/Internet uma ferramenta para a avaliação autônoma da facilidade de uso de dispositivos de software interativo. Este método tem como proposta apresentar uma técnica de avaliação, para apoiar a inspeção da interface e descobrir os defeitos ergonômicos. O projeto *ErgoList* resulta da colaboração entre o SoftPolis, núcleo Softex-2000 de Florianópolis, e o LabUtil, Laboratório de Utilizabilidade UFSC/SENAI-SC/CTAI[LAB07]. Surgiu baseado no crescente interesse das pessoas por melhorias na facilidade de uso e na utilidade dos programas de *software* interativo.

É composto por três módulos: *Checklists*, Questões e Recomendações.

- *Checklist* – Realiza uma inspeção sistemática da qualidade ergonômica da interface com o usuário do *software*;
- Questões – Permite o conhecimento das questões que compõem o módulo *Checklist*, com uma abordagem informal;

- Recomendações – Apresenta sugestões ergonômicas para o projeto de interfaces com o usuário.

Os 18 critérios elementares, aos quais estão associados os *Checklists* do *ErgoList* são:

1. Presteza,
2. Agrupamento por Localização,
3. Agrupamento por Formato,
4. Feedback,
5. Legibilidade,
6. Concisão,
7. Ações Mínimas,
8. Densidade Informacional,
9. Ações Explícitas,
10. Controle do Usuário,
11. Flexibilidade,
12. Experiência do Usuário,
13. Proteção contra erros,
14. Mensagens de Erro,
15. Correção de Erros,
16. Consistência,
17. Significados e
18. Compatibilidade.

Tab. 5.1: *Checklists* que compõem o método *ErgoList*

Lista de Avaliações <i>ErgoList</i>			
No.	Aspecto	Descrição	N. Questões
1	Presteza	Verificar se o <i>software</i> informa e conduz o usuário	17
2	Agrupamento por localização	Verificar se a distribuição espacial dos itens traduz as relações entre as informações	11
3	Agrupamento por formato	Verificar o formato dos itens como meio de transmitir associações e diferenças	17
4	Feedback	Avaliar a qualidade do feedback imediato às ações do usuário	12
5	Legibilidade	Verificar a legibilidade das informações apresentadas nas telas do <i>software</i>	27
6	Concisão	Verifica o tamanho dos diálogos e termos apresentados e introduzidos no <i>software</i>	14
7	Ações Mínimas	Verificar a extensão dos diálogos estabelecidos para a realização dos objetivos do usuário	5
8	Densidade Informacional	Avaliar a densidade informacional das telas apresentadas pelo <i>software</i>	9
9	Ações Explícitas	Verificar se é o usuário quem comanda explicitamente as ações	4
10	Controle do Usuário	Avaliar as possibilidades do usuário controlar o encadeamento e a realização das ações	4
11	Flexibilidade	Verifica se o <i>software</i> permite personalizar as telas	3
12	Experiência do Usuário	Avaliar se usuários com diferentes níveis de experiência têm iguais possibilidades de obter sucesso em seus objetos	6
13	Proteção contra erros	Verificar se o <i>software</i> oferece as oportunidades para o usuário prevenir eventuais erros	7
14	Mensagem de Erro	Avaliar a qualidade das mensagens de erro enviadas aos usuários em dificuldades	9
15	Correção de Erros	Verificar as facilidades oferecidas para que o usuário possa corrigir os erros cometidos	5
16	Consistência	Avaliar se é mantida uma coerência no projeto de códigos, telas e diálogos com o usuário	11
17	Significados	Avaliar se os códigos e denominações são claros e significativos	12
18	Compatibilidade	Verificar a compatibilidade do <i>software</i> com as expectativas e necessidades do usuário em sua tarefa	21
TOTAL			194

Cada um destes é especializado em um aspecto ou critério que determina a ergonomia de uma interface homem-computador. Além disso, cada *Checklist* possui instruções de como devem ser aplicados. As alternativas de respostas disponíveis são: Sim, Não, Não Aplicável e Adiar Resposta.

O Laudo Final é composto de:

- Total de questões,
- Total de questões respondidas,
- Total de questões não respondidas,
- Questões conformes (resposta sim),
- Questões não conformes (resposta não),
- Questões não aplicáveis e
- Questões adiadas pelo avaliador.

Técnica de Avaliação por Listas de Verificação

Este processo de avaliação é composto por dois módulos: O manual do processo de avaliação e as listas de verificação.

O Manual do processo de avaliação:

- Estratégia de Aplicação da Técnica;
- Casos de Uso;
- Guia de Questões;
- Glossário de Termos.

As listas de verificação são divididas pelos componentes:

- Relação de questões;
- Componentes Estáticos e
- Componentes Dinâmicos.

Tab. 5.2: Componentes da Técnica de Avaliação Usabilidade *software* CPqD LabIUtil

Lista de Avaliações CPqD LabIUtil		
No.	Descrição	N. Questões
1	<i>software</i> de Menu	44
2	Estrutura de formulários	51
3	Apresentação	38
4	<i>software</i> de ajuda	10
5	Diálogo	21
6	Diálogo de ajuda	04
7	Diálogo de Exceção	11
TOTAL		179

MEDE-PROS

Desenvolvido pelo CTI, Centro de Tecnologia da Informação Renato Archer, o MEDE-PROS, Método de Avaliação de Qualidade de *software* é um método utilizado para avaliar o nível de qualidade de um *software* do ponto de vista do usuário final. Foi desenvolvido baseado em Normas Internacionais ISO/IEC 9126 [ISO00, ISO91], e na ISO/IEC 12119 [ISO98b] que apresenta os requisitos de qualidade para programas de computadores.

A exemplo de outros métodos o MEDE-PROS é composto por um *Checklist*, um Manual do Avaliador e um modelo de Relatório de Avaliação. Com base na ISO/IEC 14598-5 [ISO98c] fornece recomendações para a implementação de um processo de avaliação, definindo atividades entre o solicitante da avaliação e o avaliador.

Essa avaliação é realizada simulando-se o uso normal do *software* e a utilização do mesmo de forma completa segundo sua documentação e instruções de uso. O avaliador atribui notas às questões do *Checklist* e descreve comentários que juntamente com o relatório de avaliação, vão identificar os aspectos positivos e os que devem ser revistos. Além disso descreve as sugestões para melhoria do *software*.

Questionários para avaliação de soluções WEB

Um questionário para Avaliação de *softwares* acessado via Web, deve conter questões distribuídas através de componentes de avaliação que considerem inclusive, velocidade de transmissão das informações, navegabilidade, qualidade das imagens e

Tab. 5.3: Componentes para Avaliação Usabilidade *software* WEB

Lista de Avaliações WEB		
No.	Descrição	N. Questões
1	Tempo de Carga	03
2	Aparência	14
3	Estrutura e Navegação	15
4	Conteúdo	12
5	Usabilidade	10
6	Diálogo de ajuda	04
7	Objetivos gerais de <i>Design</i>	06
TOTAL		64

das informações.

Método Heurístico para avaliação de software em EAD

O método de avaliação heurística foi proposto por Nielsen [Nie95], em colaboração com Rolf Molich em 1990, como uma das técnicas de engenharia da usabilidade, com as vantagens de baixo custo e técnicas de usabilidade consideradas mais simples para avaliação de projetos de *software*. Contando com um pequeno número de especialistas (de 3 a 5) em interface que inspecionam um protótipo procurando identificar problemas, levando em consideração as dez heurísticas ou princípios de usabilidade apresentadas a seguir:

- visibilidade, mantendo os usuários sempre informados no momento apropriado;
- usar linguagem de comunicação e a sintaxe conhecida pelo usuário;
- permitir a recuperação de erros de operação pelo usuário (undo e redo);
- ser consistente, utilizando termos, comandos e comportamentos padrões;
- privilegiar a prevenção de erros sobre a clareza de comunicação;

- navegação inteligente, minimizando o uso da memória dos usuários;
- atender eficientemente usuários experientes ou não;
- fornecer informações rápidas, claras e pertinentes;
- comunicar claramente a ocorrência de erros, de recuperação e diagnóstico;
- documentação rápida (*Help*) ou detalhada de forma fácil, objetiva e clara;

Winckler verificou [Win02] o aumento quantitativo e qualitativo de informações sobre usabilidade quando os especialistas, em um domínio de conhecimento são treinados para o testes de avaliação heurística. Observou também que cada teste cobre porções distintas de problemas de usabilidade com a interface, e devem ser utilizados em conjunto para uma avaliação mais eficiente. Ainda segundo Winckler [Win02] e também Nielsen [Nie95], o número esperado de problemas identificados por consultores treinados para o teste de avaliação heurística é significativamente maior que os verificados apenas pelos consultores e equipe sem nenhum treinamento. Observa-se que cada teste cobre problemas específicos com a interface e devem ser utilizados em conjunto para uma avaliação mais eficiente. Pode-se afirmar também que os testes, podem ser realizados tanto para solucionar problemas de usabilidade da interface, como para coordenar a equipe multidisciplinar para promover uma solução mais eficaz. Ainda numa análise crítica da utilização do método de avaliação heurística observa-se:

- Facilidade de aprendizado, pois demanda pequeno esforço de treinamento dos consultores de um domínio específico de conhecimento fora da área de informática. Isto pode também ser explicado pela natureza crítica e de inspeção do método;
- Facilidade de adaptação, permitindo que os avaliadores possam criar heurísticas específicas do domínio;
- Facilidade no encaminhamento de soluções para os problemas, utilizando os comentários, as restrições do uso de tais soluções, dando margem a um melhor aproveitamento dos conhecimentos do profissional avaliador;

- Liberdade de explanação e de investigação pelo avaliador, podendo aplicar todo o seu conhecimento sobre o problema em si. As restrições são discutidas em uma segunda etapa;
- A classificação dos problemas dá uma idéia clara de prioridade de solução;
- Otimização do tempo de participação dos consultores no projeto.

Principais métodos de avaliação, segundo Nielsen

Mais informações relativas aos principais métodos de avaliação de Usabilidade podem ser consultadas no Apêndice anexado a esta dissertação.

Métodos mais conhecidos:

- A** Aprendizagem Exploratória;
- B** Teste de Consistência;
- C** Teste de Especificações;
- D** Heurístico;
- E** Heurístico Estimado;
- F** Avaliação por uma equipe multidisciplinar;
- G** Teste Padrão;
- H** Teste Formal.

Escolheu-se para avaliação o método AvalUWeb (Método de Avaliação de Ambiente Web) com adaptações ao ambiente educacional, que será denominado AvalUEAD (Avaliação de Usabilidade para Ambientes de Educação a Distância). Este se mostrou adequado também para o acompanhamento de todo o ciclo de vida do LVCCA (Laboratório Virtual para Circuitos de Corrente Alternada). Fator muito importante, já que se trata de uma ferramenta educacional cujo desenvolvimento permanente é um de seus requisitos principais, com a conseqüente inclusão de novas funcionalidades, inclusive por autores diferentes.

5.1 Método AvalUEAD

O método AvalUEAD, desenvolvido para este trabalho, será utilizado para avaliação de usabilidade de *softwares* educacionais em ambientes WEB.

Trata-se de uma contribuição ao método denominado AvalUWeb, desenvolvido por Priscilla de Barros Basso Pagliuso, em seu trabalho de Mestrado [Pag04] sob orientação do Professor Doutor Ademir José Petenate do Instituto de Matemática, Estatística e Computação Científica-IMECC UNICAMP.

Foi elaborado considerando os princípios de usabilidade apresentados por Jakob Nielsen [Nie95], pela série de Normas ISO 9241(ISO 9241-1, 1997)(ISO 9241-10, 1993)(ISO 9241-11, 1998) e pela Norma ISO/IEC 9126-1(ISO/IEC 9126-1, 2000).

Este método tem o objetivo de ajudar na elaboração e principalmente na avaliação de *softwares* educacionais a distância, aprimorando a qualidade durante todo o processo de desenvolvimento e utilização do sistema.

5.1.1 Características do Método

O AvalUEAD, um método de Avaliação de Interfaces Educacionais, é baseado nos princípios de usabilidade e foi desenvolvido utilizando o conceito PDCA (Planejamento, Execução, Verificação e Ação Corretiva) [Pes01]. Passando pelas fases de definição dos objetivos e métodos, construção, verificação dos resultados através do estudo de caso e ações corretivas de melhorias obtidas após os estudos de casos identificados no projeto LVCCA (Laboratório Virtual para Circuitos de Corrente Alternada).

São três os principais atores envolvidos durante todo o processo de utilização do método:

1. Avaliador, responsável por realizar a avaliação;
2. Coordenador da avaliação, coordena o(s) avaliador(es), prepara a documentação durante a avaliação e interage com o patrocinador;
3. Patrocinador, financiador responsável pelo projeto.

Este método pode de ser utilizado durante o desenvolvimento ou então na fase de avaliação do *software*. Característica dos conhecidos ciclos de melhoria contínua da metodologia PDCA [Pes01].

Neste caso, os papéis dos Avaliadores e do Coordenador foram exercidos por profissionais com afinidades e habilidades relacionadas com as características do

software desenvolvido. Utilizou-se este método como instrumento para medir a qualidade das interfaces do *software*, baseado nos princípios de usabilidade e ergonomia propostos através dos 13 componentes de avaliação que compõem o *Checklist*, além da padronização dos resultados da avaliação através dos relatórios apresentados pelo método AvalUEAD. De forma a atender itens específicos inerentes a uma ferramenta educacional, foram efetuadas adaptações na quantidade e nas descrições das questões.

A medição realizada através deste método é composta por:

Checklist : utilizado pelo avaliador como um guia durante o processo de avaliação.

Este *Checklist* é composto por 13 componentes e um total de 131 questões, distribuídas através de componentes de avaliação, como descrito na tabela 5.4;

Relatório de Avaliação : utilizado individualmente pelo(s) avaliador(es) durante o processo de avaliação e utilização do *Checklist*;

Relatório de Resultados : contendo a tabulação dos resultados do Processo de Avaliação é utilizado pelo coordenador da avaliação após a conclusão do processo de avaliação pelos avaliadores para tabular os resultados de todos os avaliador(es) individualmente, com suas notas e sugestões;

Laudo Final da Avaliação : utilizado pelo coordenador da avaliação após a conclusão do processo de avaliação pelo(s) avaliador(es). Este laudo apresenta ao patrocinador do site os resultados obtidos com a avaliação e as sugestões de melhorias.

Checklist para avaliação de interfaces, segundo princípios de usabilidade

Os 13 componentes elementares do Método AvalUEAD são:

1. Apresentação do *software* Avaliado: é a primeira impressão do avaliador, portanto deve ser avaliada após uma breve navegação. Com questões gerais sobre o *software* avaliado também aborda itens relativos ao desenvolvimento, sem se prender a muitos detalhes;
2. Estrutura do *software*: verifica a estrutura do *software* em termos de organização e seqüência de telas e comandos para que haja uma boa navegação;
3. Funcionalidade das Telas: Uma vez avaliadas a Apresentação e a Estruturação do *software*, avalia-se então os objetivos de cada função apresentada nas telas.

Tab. 5.4: Questionário de Avaliação de Usabilidade de Soluções EAD

Lista de Avaliações de <i>software</i> EAD		
No.	Descrição	N. Questões
1	Apresentação	18
2	Estrutura do <i>software</i>	03
3	Funcionalidade	20
4	<i>Links</i> e Controles	11
5	Atratividade	09
6	Varredura	08
7	Sub-telas e Janelas	02
8	Interoperabilidade EAD	10
9	Legibilidade	06
10	Documentação <i>On-line</i>	07
11	Multimídia	08
12	Distância Transacional	07
13	Didática e Cognição	09
TOTAL		118

Se estão bem definidas de acordo com os padrões usuais do ambiente, verificam-se as informações relacionadas à navegação, a localização dos menus em uma área de fácil acesso e a definição de comandos.

Verifica-se também a comunicação das telas com o usuário, se são amigáveis, se são informações e se os recursos utilizados são facilmente localizados pelos usuários;

4. Links e Controles: verifica-se as funções dos *links* e controles, itens de extrema importância em um hipertexto, responsáveis por fazer a ligação entre os diversos comandos e ícones no *software*;
5. Atratividade: importante para a motivação de alunos e professores, avalia-se o uso de telas coloridas, padronizadas e organizadas, com recursos inteligentes. Verifica-se também a distribuição das informações e a qualidade do acabamento das interfaces, imagens e efeitos especiais. O tema escolhido para as telas, os detalhes, a combinação de cores e sua afinidade com a proposta educativa do *software*;
6. Varredura: em *softwares* utilizados para educação é necessário assegurar principalmente, que haja aderência às regras gramaticais. Desta forma são levantadas questões que consigam identificar a existência de possíveis falhas;
7. Sub-telas e Janelas: verifica-se a utilização das sub-telas e janelas especiais como um recurso muito ilustrativo, mas que precisa ser implementado considerando-se um bom efeito de apresentação, o tamanho adequado, o conteúdo, a forma e o momento oportuno;
8. Interoperabilidade EAD: considerando-se a inevitável utilização da Internet como meio de acesso alternativo à informação, e na busca pelo conhecimento, este item foi adaptado à medição da facilidade de integração do *software* a outros *softwares* operacionais e a outros ambientes de EAD;
9. Legibilidade: verifica se as telas são legíveis pois, caso o usuário não possa ler o texto da tela, questões sobre *Design*, velocidade e conteúdo do site passam a ser secundários. É interessante acrescentar que a leitura na tela do micro é 25% mais lenta que a leitura feita em papel, portanto letras muito pequenas devem ser reservadas para notas de rodapé. Textos com “CAPS LOCK” tornam a leitura 10% mais lenta [Nie95];
10. Documentação On-line: deve ser utilizada quando o site tem um *Design* sofisticado sendo necessária uma documentação vasta e completa, embora usuários

normalmente sejam impacientes para ler documentação e só a procurem em caso de problema ou dificuldade na utilização;

11. Multimídia: relativo a apresentação dos recursos de animação, vídeo e áudio que devem ser usados com muito cuidado pois além de recursos lentos podem sobrecarregar o usuário com excesso de movimento na tela;
12. Distância Transacional: outro item modificado e adaptado ao ambiente educacional, avalia o nível de interação entre os alunos, os educadores, os colegas e os demais recursos do ambiente;
13. Didática e Cognição: também adaptado a soluções educacionais, avalia a capacidade de transmitir o conhecimento e o nível de assimilação pelos usuários do *software*.

O *Checklist* deve ser respondido aplicando-se os testes ao ambiente definido nas fases de análise e especificação preliminar, utilizando-se para isso a identificação das características da interface do *software* em avaliação. Os avaliadores também devem ser experientes em desenvolvimento de *softwares* relacionados com as características do sistema a ser avaliado e que tenham também, realizado o treinamento inicial de como utilizar o método.

Conforme descrito anteriormente, o *Checklist* é dividido por componentes e que apresentam um conjunto de questões que devem ser avaliadas na utilização do *software*. As questões por sua vez apresentam 3 (três) opções de respostas: Sim (notas de 6 a 10), Não (notas de 0 a 4), e Não Aplicável (5).

O método prevê que os avaliadores respondam todas as questões em sequência e a resposta dada a cada questão será individualmente analisada posteriormente.

Qualquer anormalidade que prejudique o processo de avaliação deve ser apontado no campo “Comentários e Observações” apresentado ao final de cada componente. Nesse mesmo campo, devem ser apresentadas informações sobre qualquer outro problema ou sugestão que se fizerem necessárias, relativas a cada componente.

As observações e resultados precisam ser consistentes, ou seja, que possam ser reproduzidos nas mesmas condições de teste por outro avaliador. Para isso, o *Checklist* deve atender a princípios como organização, clareza e simplicidade, contribuindo desta forma, para repetibilidade e reprodutibilidade do método.

O perfil homogêneo do avaliador contribui para a parametrização dos resultados e para a reprodutibilidade.

5.2 Avaliação do Sistema pelo AvalUEAD

Tab. 5.5: Apresentação do *software*

Apresentação	
No.	Avaliação
1	De maneira geral, as páginas apresentam cores e efeitos sem excesso?
2	A página apresenta um número adequado de figuras para integrar o usuário ao que está sendo proposto? a- Isto está sendo feito de maneira rápida e eficaz?
3	O <i>software</i> contém informações suficientemente detalhadas e rápidas?
4	O <i>software</i> contém informações precisas, relevantes e atualizadas?
5	O <i>software</i> está livre de informações confusas e incompletas?
6	A navegabilidade é feita com o máximo de objetividade?
7	A seção “Fale com o Autor” está em lugar bem visível no <i>software</i> ? a- E o fone, fax e endereço real?
8	O “mapa do <i>software</i> ” está em lugar visível?
9	O <i>software</i> possui a opção “Search” ou “Localizar” visível?
10	O <i>software</i> permite a Interação com outros usuários? (Chats, Mensagens, Atividades em Grupo...)
11	Os comandos mais utilizados (mais frequentes, mais acessados) têm lugar de destaque no <i>software</i> ?
12	A <i>software</i> possui um layout intuitivo e dinâmico?
13	O <i>software</i> possui um resumo com as informações relevantes e um link para maiores detalhes?
14	Existem ícones ou imagens de acesso rápido?
15	Existe efeito de animação? a- Estes são apresentados de maneira rápida e adequada?
16	Existe uma opção (<i>Help</i> , e-mail) para o usuário tirar suas dúvidas sobre o <i>software</i> ? a- Essa opção é fácil de ser encontrada e utilizada? b- Em caso de e-mail, o tempo de resposta ao usuário/cliente é inferior a 2 horas?
17	O <i>software</i> está livre do uso de metáfora em suas telas?
18	O <i>software</i> superou as suas expectativas?

Tab. 5.6: Estrutura do *software*

Estrutura	
No.	Avaliação
1	A estrutura do <i>software</i> é coerente em cada atividade, a base de informações, a de dados e de ferramentas estão organizadas, agrupadas e de fácil utilização?
2	A estrutura do <i>software</i> apresenta informações lineares, seqüenciais, segue uma lógica de progressão de uso?
3	O <i>software</i> apresenta informações bem organizadas e sob uma orientação pedagógica?

Tab. 5.7: Funcionalidade do *software*

Funcionalidade	
No.	Avaliação
1	A navegação do <i>software</i> é rápida e fácil?
2	Os menus estão localizados em uma área de acesso fácil? a- Os menus são de fácil navegação?
3	O <i>software</i> apresenta um botão “Home” de fácil acesso? a- O <i>software</i> possui um link para um botão “Home” em todas as situações?
4	É possível usar recursos para aproximar ou afastar imagem (Zoom), navegar horizontal e verticalmente na tela?
5	Os principais recursos do <i>software</i> são facilmente acessados?
6	Durante a utilização do <i>software</i> , é possível progredir e retroceder com rapidez e facilidade?
7	O <i>software</i> está livre de informações desnecessárias e relacionadas com a área de interesse?
8	O <i>software</i> permite desativar mensagens e alertas conhecidos?
9	O <i>software</i> permite que seja salvo e recuperado determinado nível de utilização ou exercício para que seja continuado posteriormente?
10	Em caso do <i>software</i> possuir algum tipo de registro ou estatística, estes são simples, com questões selecionadas e informações relevantes?
11	Existe um vínculo de relacionamento do <i>software</i> com o cliente?
12	O <i>software</i> permite o cadastro e manutenção de grupos de utilização?
13	O <i>software</i> apresenta links para outras fontes de referência ou informações adicionais?
14	É possível selecionar a informação desejada em menos de três (3) cliques?
15	É possível utilizar o <i>software</i> através de plataformas simples de hardware e <i>software</i> ?
16	É possível encontrar o <i>software</i> como parte de outros sistemas educacionais?
17	O <i>software</i> está livre de situações sem saída, sem alternativa?
18	O <i>software</i> apresenta um descritivo das próximas versões e dos planos futuros?
19	O <i>software</i> apresenta de forma clara quais os procedimentos que garantem a segurança das informações?
20	O <i>software</i> está livre de erros/falhas como queda do <i>software</i> no meio das lições? a- Caso exista algum tipo de falha é apresentado um aviso que permita facilmente a identificação do problema ocorrido?

Tab. 5.8: Links e controles do *software*

Links e controles	
No.	Avaliação
1	É visível a apresentação de links e controles?
2	O <i>software</i> apresenta distinção de duas cores diferentes nas seleções, para diferenciar quais foram acessadas e não acessadas, facilmente percebida pelo usuário?
3	Todo link, não óbvio, possui o título do link?
4	Caso o link vá para um <i>software</i> diferente está claramente especificado no título do link, o nome do <i>software</i> que será linkado?
5	Caso o link ou controle acione um <i>subsoftware</i> , está claramente especificado no título do link, o nome do <i>subsoftware</i> que será linkado?
6	O título do link ou controle apresenta uma explicação rápida e geral sobre o que poderá ser encontrado ao acessar este link?
7	O título do link ou controle apresenta aviso sobre possíveis dificuldades encontradas ao acessar este link?
8	O título do link possui menos de 60 caracteres?
9	É utilizada sempre a mesma URL para o <i>software</i> ?
10	É apresentado no <i>software</i> um link para registro do usuário?
11	O <i>software</i> apresenta Advertising links?

Tab. 5.9: Atratividade do *software*

Atratividade	
No.	Avaliação
1	O <i>software</i> apresenta um mesmo estilo de folha para todas as suas páginas?
2	É possível encontrar no <i>software</i> um pequeno manual on-line explicando os diferentes estilos apresentados, caso isso ocorra?
3	O <i>software</i> apresenta telas ou páginas em estilo?
4	O <i>software</i> apresenta menos de duas fontes de texto?
5	Caso o <i>software</i> apresente uma terceira fonte de texto, ela está sendo usada para apresentar um texto de código computacional?
6	A fonte apresenta tamanho relativo dependendo do programa utilizado ou do tamanho da janela?
7	O “ <i>search</i> ” ou “ <i>procura</i> ” está livre de resultados repetidos/idênticos?
8	A utilização de cores é agradável?
9	Existem muitos recursos de animação, eles são claros e de boa qualidade educacional?

Tab. 5.10: Varredura do *software*

Varredura	
No.	Avaliação
1	O <i>software</i> apresenta informações sucintas?
2	O <i>software</i> apresenta parágrafos pequenos, lista de marcadores (<i>bullets</i>) e palavras sublinhadas para facilitar a leitura do usuário?
3	O <i>software</i> apresenta link com hipertexto para explicar e apresentar longas informações?
4	O <i>software</i> apresenta na primeira página cabeçalho atraente, confortável e inteligente para atrair o usuário num primeiro instante?
5	Os textos do <i>software</i> estão estruturados para facilitar a visibilidade e leitura?
6	O texto do <i>software</i> apresenta marcadores (<i>bullets</i>) para facilitar a leitura e dar quebras aos textos uniformes?
7	O <i>software</i> utiliza <i>highlights</i> e cores para dar ênfase às palavras mais importantes no texto?
8	O efeito <i>highlights</i> está sendo usado com cautela?

Tab. 5.11: Sub-telas e Janelas do *software*

Sub-telas e Janelas	
No.	Avaliação
1	Em caso de utilização de Sub-telas ou Janelas é possível ter uma boa visualização das informações na tela?
2	Utilizando-se Sub-telas ou Janelas é possível obter uma boa navegação?

Tab. 5.12: Interoperabilidade EAD do *software*

Interoperabilidade EAD	
No.	Avaliação
1	As telas apresentam um título ou sinalização que permita a identificação deste <i>software</i> quando sendo utilizado como parte de um outro <i>software</i> educacional?
2	É possível inserir facilmente este <i>software</i> em outras plataformas ou <i>softwares</i> educacionais?
3	É possível identificar com clareza o conteúdo exato do <i>software</i> somente pelo título utilizado?
4	É apresentado um texto de 40 a 60 caracteres, além do título da página, para explicar o que o usuário poderá fazer com este <i>software</i> ?
5	Se o usuário adicionar o <i>software</i> em seu histórico o título da página utilizado é capaz de especificar exatamente o que é o <i>software</i> ?
6	O Título da página está livre de artigos como A, O, Esse ou Essa que limita o número de caracteres?
7	O <i>software</i> está livre de informações negativas? (Ex: o <i>software</i> não apresenta ...)
8	O <i>software</i> permite o ajuste automático de escala de suas telas quando inserido em outros ambientes como sub-tela? (Como <i>FRAME</i> de uma lição do Teleduc, por exemplo)
9	É possível utilizá-lo facilmente em vários ambientes operacionais, como Windows, Linux e outros ?
10	O <i>software</i> quando adaptado a outros ambientes ou sistemas operacionais necessita de algum dispositivo adicional de <i>software</i> ou hardware dedicado?

Tab. 5.13: Legibilidade do *software*

Legibilidade	
No.	Avaliação
1	O <i>software</i> apresenta texto positivo, isto é, texto escuro com background claro?
2	As letras dos textos apresentadas na página são de grande contraste com o fundo da tela?
3	As letras são visíveis, isto é, apresentam um bom tamanho facilitando assim a leitura?
4	O <i>software</i> apresenta textos alinhados a esquerda facilitando a leitura?
5	O <i>software</i> apresenta textos centralizados e alinhados a direita para dar efeito e atrair a atenção?
6	A página está livre de texto com formato Caps Lock?

Tab. 5.14: Documentação *on-line* do *software*

Documentação <i>on-line</i>	
No.	Avaliação
1	O <i>software</i> apresenta documentação ou manual <i>on-line</i> ?
2	O <i>software</i> apresenta <i>Help on-line</i> ?
3	Toda a documentação existente em papel está disponível na forma <i>on-line</i> ?
4	A documentação <i>on-line</i> está visível e fácil de ser encontrada?
5	A documentação <i>on-line</i> apresenta um <i>step-by-step</i> ?
6	A documentação <i>on-line</i> apresenta exemplos em abundância?
7	É apresentado um glossário com os termos utilizados nas telas?

Tab. 5.15: Recursos Multimídia

Multimídia	
No.	Avaliação
1	Caso o <i>software</i> apresente algum arquivo para download é apresentada ao usuário a informação exata do que é o arquivo e para que serve o download que será feito? a- É apresentado entre parênteses o tamanho do arquivo? b- É apresentado o tempo que levará o download completo e a velocidade que esse download será/está sendo feito?
2	Caso haja um link com um arquivo para download (.pdf por ex.), que necessite de um programa especial (Adobe Acrobat, por ex.) para abertura é apresentado como, onde e/ou um link com instruções de como baixar o programa?
3	A página apresenta recursos de foto e/ou texto explicativo sobre o que irá ser aberto (ver ou ouvir)? a- Caso apresente recursos de foto, esta é apresentada em tamanho pequeno? b- Caso apresente recursos de foto, existe um <i>link</i> para ampliar e visualizar melhor a foto?
4	A animação está livre de loop infinito?
5	Caso a tela apresente animação, é apresentada uma opção ao usuário de cancelar ou pular a animação?
6	As telas estão livres de efeitos desnecessários que poluem visualmente e causam atrasos na execução?
7	Caso a página apresente vários objetos de informações em um mesmo espaço, os objetos se movem apenas quando apropriado ou quando o cursor é posicionado sobre a imagem?
8	Em caso de biografia, é apresentada a foto da pessoa?

Tab. 5.16: Distância Transacional

Distância Transacional	
No.	Avaliação
1	Existem recursos para atividades em grupo?
2	Existe a figura do Educador ou seu representante, para apoio durante as lições? (mesmo que as respostas não sejam <i>on-line</i>)
3	O <i>software</i> possui ou possibilita a comunicação entre os alunos? (mesmo que esta comunicação ocorra através da cooperação com outros <i>softwares</i>)
4	O <i>software</i> promove a comunicação entre os alunos referente a outros assuntos, como ferramenta de integração? (mesmo que esta comunicação ocorra através da cooperação com outros <i>softwares</i>)
5	O <i>software</i> possibilita o cadastramento do perfil dos demais alunos, como ferramenta de integração? (mesmo que esta comunicação ocorra através da cooperação com outros <i>softwares</i>)
6	Existe um eficiente sistema de avaliação e de acompanhamento da frequência de participação do aluno? (mesmo que este controle ocorra através da cooperação com outros <i>softwares</i>)
7	O <i>software</i> permite o desenvolvimento individual, como promotor da iniciativa e da criatividade?

Tab. 5.17: Didática e Cognição

Didática e Cognição	
No.	Avaliação
1	O Software abrange todo o conteúdo teórico a que se propõe?
2	As lições são apresentadas segundo uma orientação crescente do volume de conhecimento adquirido?
3	Existe um procedimento recorrente de aprendizagem, avaliação e correção?
4	É permitido ao aluno desenvolver, por si próprio, situações e cenários referentes ao tema abordado?
5	As lições conseguem transmitir as informações de maneira clara e objetiva?
6	Os recursos multimídia, são utilizados com qualidade e objetividade de modo a facilitar a assimilação do conhecimento que se deseja transmitir?
7	É previsto a interação com outros <i>softwares</i> educacionais e com outros grupos de estudo afins?

5.2.1 Relatório Final de Avaliação

Esse relatório deverá ser preenchido pelo(s) avaliador(es) durante a avaliação, em paralelo com o *Checklist* e após a conclusão, ambos deverão ser entregues ao coordenador contendo: A quantidade de horas utilizadas pelo avaliador para avaliar o *software*, a descrição dos erros encontrados no *software* através da avaliação e as observações gerais com relação a erros encontrados por componente.

Assim sendo, a quantidade de horas utilizadas pelo avaliador para realizar a avaliação servirá como um histórico para planejamento do tempo necessário para as avaliações futuras garantindo o máximo de produtividade por tempo de avaliação. Os erros encontrados no *software* através da avaliação devem ser relatados apresentando detalhes de como o avaliador chegou até o erro, quais foram os últimos passos para que o erro tenha ocorrido e o tipo de falha ocorrida (ex: arquivo não encontrado ou problemas com conexão). As observações gerais com relação aos erros encontrados por componente, erros ortográficos, sugerindo alterações ou até mesmo notificando a falta de compreensão de alguma questão apresentada, deverão ser descritas.

Após a entrega do relatório pelo(s) avaliador(es), o coordenador da avaliação sintetiza as informações contidas em um ou mais relatórios, conforme o número de avaliadores, e detalha estas informações no Relatório de Resultados.

Tab. 5.18: AVALIADOR 1 – Relatório Individual de Avaliação

AVALIADOR 1 – Relatório Individual de Avaliação			
Coordenador	COORDENADOR 01		Data Final 27/08/2008
Avaliador	AVALIADOR 01		Total Horas 2h 57m
Software Avaliado	Laboratório Virtual de Corrente Alternada		
Principais Dificuldades Encontradas	Os testes foram submetidos ao Laboratório Virtual considerando-se a existência de um software hospedeiro, como o Teleduc ou o Moodle por exemplo, para tratar etapas importantes de Usabilidade. Os requisitos de velocidade, portabilidade, leveza e flexibilidade são contraditórios com outras especificações técnicas como códigos otimizados, efeitos e imagens de alta definição.		
Observações Gerais			
Cálculo Hora			
Data	Hora Inicial	Hora Final	Descrição
27/08	9h20m	12h17m	Tempo medido para avaliação de todos os itens
Principais Falhas e Erros			
No.	Descrição		
01	Dificuldade na seleção de itens muito pequenos como os terminais dos mediadores, principalmente quando estes se sobrepõem a outros componentes		

Tab. 5.19: AVALIADOR 1 – Considerações Técnicas Individuais por Componente

AVALIADOR 1 – Considerações Técnicas	
Componentes	Descrição
Apresentação	Média 6.7 – O software agrada demonstrando leveza e simplicidade de interação, principalmente com bom enquadramento e recursos em ambientes WEB.
Estrutura do <i>software</i>	Média 9.6 – Boa estrutura didática, simples, com boa organização de elementos e semelhança com ambiente real promovendo uma atividade exploratória do usuário.
Funcionalidade	Média 6.9 – A flexibilidade quanto ao hardware mínimo e ao sistemas operacionais suportados asseguram um bom desempenho funcional, apesar de não ter sido verificada alguma falha para que se pudesse avaliar a qualidade dos avisos e alertas.
Links e controles	Média 6.0 – Diversos itens de avaliação estão associados ao software hospedeiro, como o Teleduc por exemplo. O destaques positivos são a utilização de ícones e a facilidade de ampliação das imagens.
Atratividade	Média 6.8 – A harmonia das cores e estilos utilizados favorecem a avaliação apesar das restrições quanto à riqueza de animações e efeitos.
Varredura	Média 6.0 – Não foram verificados <i>links</i> com hipertextos ou de destaque. As informações complementares foram atribuídas aos softwares hospedeiros.
Sub-telas e Janelas	Média 7.0 – As instâncias dos objetos das bibliotecas de signos constituem-se nas formas de janelas independentes com boa visualização e navegabilidade.
Interoperabilidade EAD	Média 8.7 – A utilização de vários sistemas operacionais, plataformas de hardware, softwares hospedeiros e navegadores, asseguram a operabilidade consistente e coerente do software.
Legibilidade	Média 6.8 – A associação com ícones, objetos e efeitos gráficos minimizaram a utilização de textos tornando este item pouco efetivo para este tipo de avaliação associada à textos.
Documentação <i>on-line</i>	Média 2.6 – Não foram verificados, neste nível, informações “ <i>on-line</i> ”, ou mesmo “ <i>Help on-line</i> ”.
Multimídia	Média 6.9 – Os requisitos de flexibilidade e leveza restringiram o uso dos recursos multimídia a algumas modalidades de animação com menor esforço computacional.
Distância Transacional	Média 8.1 – A boa interação com o ambiente de simulação deve ser complementada através do software hospedeiro, ao qual foi atribuída a responsabilidade pela associação e integração dos usuários do software.
Didática e Cognição	Média 7.7 – O software permite ao usuário testar uma grande variedade de situações. Os recursos de animação, apesar de objetivos e simples, facilitam a simulação de montagens e configurações necessárias aos objetivos pedagógicos do software.

Tab. 5.20: AVALIADOR 2 – Relatório Individual de Avaliação

AVALIADOR 2 – Relatório Individual de Avaliação			
Coordenador	COORDENADOR 01		Data Final 27/08/2008
Avaliador	AVALIADOR 02		Total Horas 2h 08m
Software Avaliado	Laboratório Virtual de Corrente Alternada		
Principais Dificuldades Encontradas	Necessidade de um roteiro pedagógico relativo ao assunto que se deseja ensinar. Interação com outros ambientes educacionais.		
Observações Gerais			
Cálculo Hora			
Data	Hora Inicial	Hora Final	Descrição
27/08	10h10m	12h18m	O tempo foi computado em um único período
Principais Falhas e Erros			
No.	Descrição		
01	O deslocamento dos componentes não ficou restrito a área de trabalho.		

Tab. 5.21: AVALIADOR 2 – Considerações Técnicas Individuais por Componente

AVALIADOR 2 – Considerações Técnicas	
Componentes	Descrição
Apresentação	Média 7.4 – Boa apresentação verificada para testes utilizando-se elementos simples, considerada a integração com Teleduc.
Estrutura do <i>software</i>	Média 10.0 – Atendeu as expectativas com comandos simples, desafiando o usuário a associação de elementos e dispositivos em situações de aprendizado.
Funcionalidade	Média 6.7 – A existência de itens não aplicáveis concentra o tópico nas especificações de hardware e sistemas operacionais, onde o sistema se mostrou com boa acessibilidade e flexibilidade.
Links e controles	Média 6.3 – Bom desempenho nos itens relacionados aos controles, com destaque para o suporte decorrente do uso da linguagem FLASH. Os itens referentes aos links foram considerados não aplicáveis.
Atratividade	Média 6.1 – Bom estilo utilizado na geração de instâncias dos componentes e telas dos equipamentos de medição que, apesar de simples, favoreceram a leveza e a portabilidade.
Varredura	Média 8.5 – As “palavras” no método foram interpretadas como objetos. Os efeitos de destaque e os campos de seleção facilitam a utilização e interação.
Sub-telas e Janelas	Média 10.0 – Para tornar a avaliação aplicável, considerou-se as instâncias de objetos das bibliotecas, onde se verificou boa definição e facilidade de utilização.
Interoperabilidade EAD	Média 8.8 – Flexibilidade para utilização em vários ambientes educacionais sem a necessidade de utilização ou aquisição de softwares específicos.
Legibilidade	Média 8.2 – A interpretação é facilitada pelo uso de elementos gráficos simples e diretos o que facilita a leitura das informações transmitidas.
Documentação <i>on-line</i>	Média 2.8 – A documentação on-line, pouco disponível, foi pouco explorada dada a natureza do teste e a pressuposta interação com um ambiente educacional.
Multimídia	Média 6.9 – Animações simples poderiam ser enriquecidas com recursos sonoros, apesar do comprometimento com a interoperabilidade e a flexibilidade.
Distância Transacional	Média 9.3 – O software apresenta boa interatividade. Este item foi avaliado considerando-se a utilização integrada dos recursos normalmente disponíveis em outros softwares educacionais.
Didática e Cognição	Média 8.9 – A capacidade de simular diferentes ambientes de aprendizado, favorece a interpretação dos conceitos a serem assimilados. Os testes consideram situações simples da área de eletrotécnica.

Tab. 5.22: Laudo Final de Avaliação

Laudo Final de Avaliação			
N. de Avaliadores: 2		N. de Questões: 116	
Data: 27/08/2008		Software: LVCCA.SWF	
<input type="checkbox"/> Pré-Projeto		<input checked="" type="checkbox"/> Pré-Release	
<input type="checkbox"/> Release		<input type="checkbox"/> Versão	
Avaliador	Tempo	Resultado	Observação Geral
01	2h57m	Aceitável	Alguns termos utilizados na ferramenta de avaliação ainda são de uso geral, a avaliação poderia ser ainda mais eficaz com um foco maior em itens relacionados somente com Educação a Distância, ainda em um contexto específico. Para esta fase de desenvolvimento o software atendeu os requisitos principais de portabilidade e flexibilidade para aplicações educacionais, o que é muito importante, veja o gráfico na figura 5.1.
02	2h08m	Aceitável	Uma terminologia muito específica dificultou um pouco a avaliação, são alguns termos e símbolos da área de Eletro-eletrônica ou do processo de avaliação. A ferramenta é de propósito geral e a avaliação considerou muitos itens em colaboração com outros softwares educacionais. A documentação on-line é o exemplo de um item que pode ser atendido em um nível superior da aplicação ou se mostrará deficiente neste nível, veja o gráfico na figura 5.1.
Estatísticas totais (2 avaliadores)			
Notas	N.Total	%	Parecer
(6 a 10)	138	59.5%	Destacaram-se os itens “Distância Transacional”, “Didática e Cognição” e “Interoperabilidade EAD”
(5)	68	29.3%	A porcentagem de itens “Não Aplicáveis” se deve principalmente a atribuições feitas ao software hospedeiro como o Teleduc, por exemplo.
(0 a 4)	26	11.2%	Será necessária uma melhoria na documentação on-line, conforme o gráfico figura 5.1.

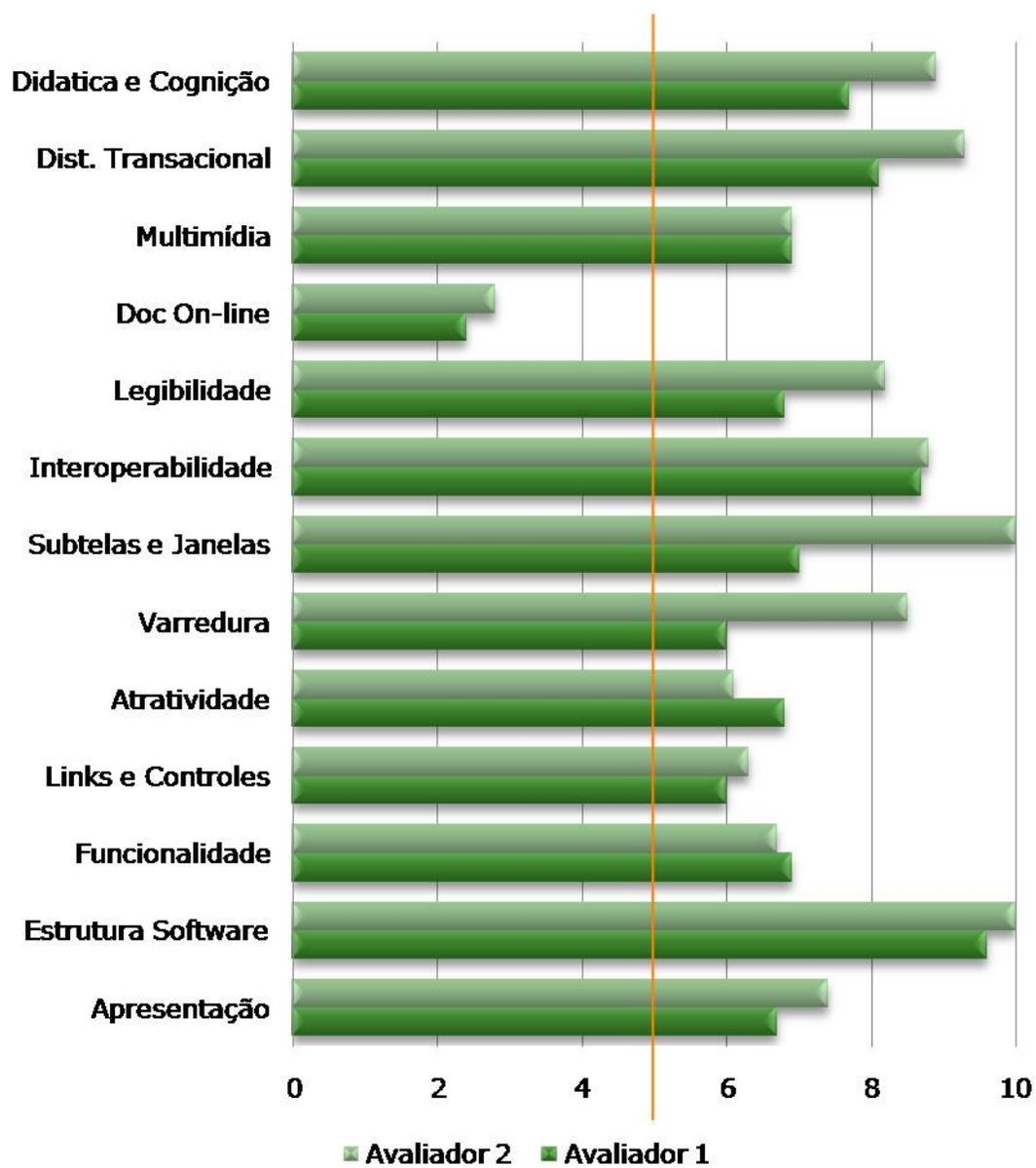


Fig. 5.1: Gráfico AvalUEAD

5.3 Conclusão deste capítulo

Neste capítulo utilizou-se vários modelos de teste de Usabilidade para fundamentar o desenvolvimento de um método específico para validação de softwares educacionais, o AvalUEAD. Este novo modelo aqui proposto levou em consideração:

- Apresentação do Software
- Estrutura
- Funcionalidade
- Links e Controles
- Atratividade
- Varredura
- Sub-telas e Janelas
- Interoperabilidade EAD
- Legibilidade
- Documentação On-line
- Recursos Multimídia
- Distância Transacional
- Didática e Cognição

Apresentou-se então o relatório Final de Avaliação juntamente com as considerações técnicas individuais por componente do método. Em resumo, o software alcançou os objetivos com um conjunto de resultados satisfatórios na maioria dos itens avaliados. Destacou-se também a importância da sistematização no desenvolvimento de software e a contribuição da engenharia ao ser aplicada na metodologia de criação e manutenção de programas de computadores. Utilizando-se dos conceitos da UML (Unified Modeling Language) efetuou-se o modelamento de classes e objetos para a implementação do núcleo do modelo proposto neste trabalho. No próximo capítulo apresenta-se o modelo tridimensional de desenvolvimento, além da arquitetura e a implementação do núcleo do Laboratório Virtual.

Capítulo 6

CONCLUSÕES

Não se pode negar o uso crescente das ferramentas virtuais em todas as áreas da ciência e o mesmo acontece nos cursos de Engenharia. Os recursos tecnológicos têm aprimorado esta alternativa, acelerando os estudos e o desenvolvimento de novas ferramentas, seja para a geração de modelos e protótipos ou nas simulações e ensaios de teste. Em paralelo, os conceitos pedagógicos e o estudo do conhecimento tem auxiliado a introdução destas ferramentas quando aplicadas à educação. A Semiótica, em especial, pode promover uma mudança no desenvolvimento deste tipo de solução, aumentando a eficiência do ensino através de novas metodologias. Em adição, ambientes educacionais como o Teleduc, Moodle e outros, podem ajudar a gerenciar a distância transacional que é outro indicador utilizado para qualificar a eficiência do aprendizado e está relacionada com o número de interações produtivas entre os envolvidos no processo de aprendizagem, incluindo os estudantes, os professores e os recursos pedagógicos disponíveis.

Não obstante, cabe ressaltar que, os recursos tecnológicos podem ser extremamente diferentes de um ambiente para outro, das especificações dos computadores até a capacidade das redes, se disponível. Desse ponto de vista a escolha do *software* apropriado e de suas funcionalidades podem ser fundamentais para o sucesso do projeto de um laboratório virtual.

A proposta de um desenvolvimento em camadas segundo uma implementação 3D, pode auxiliar no processo cognitivo e ainda determinar o limite para o uso de ferramentas e recursos de modo a preservar a eficiência da solução. Sendo assim, o Núcleo do Sistema Virtual foi projetado para trabalhar mesmo em pequenas plataformas utilizando *softwares* de navegação populares com “*FLASH plug-ins*”. Uma implementação modular pode resolver muitas dificuldades e preservar a usabilidade da solução, com um bom compromisso entre as camadas desenvolvidas, os recursos, a qualidade e o desempenho. O futuro da realidade virtual, dos objetos de aprendizagem e do controle a distância podem trazer grandes possibilidades para esta ferramenta, principalmente dentro dos cursos de Engenharia.

Capítulo 7

TRABALHOS FUTUROS

Como sugestões para trabalhos futuros propõe-se:

- Desenvolver conexões e cenários mais complexos entre os componentes, utilizando conhecimentos de Redes Neurais;
- Desenvolver a biblioteca de componentes modelados em frames VP12;
- Modelar componentes mais complexos utilizando-se *frames* com mais de 12 terminais (VPn);
- Desenvolver equipamentos de medição e monitoramento mais sofisticados como analisadores de espectro e outros;
- Testar a plataforma EAD em modelos teóricos de outras áreas da ciência e da engenharia como Química, Alimentos, Mecânica e outras;
- Implementar os demais níveis da suite 3D, especialmente o ambiente cliente/servidor para interação com o laboratório real;
- Desenvolver o PBV (Protoboard Virtual) interativo dinâmico, livre e animado que possibilite a criação ou identificação de elementos em tempo de execução;
- Desenvolver a ferramenta de desenho livre de componentes utilizando reconhecimento de padrões;

Referências Bibliográficas

- [And03] Andrade, Fábio Felipe de. O método de melhorias PDCA. Tese de mestrado, USP - Universidade de São Paulo, Novembro 2003.
- [Ara00] Araújo, Eratóstenes E. R.; Meira, Silvio R. L. SOFTEX Inserção Competitiva do Brasil no Mercado Internacional de Software. *SOFTEX*, Dezembro 2000.
- [Ara04] Aranha, Gláucio. O processo de consolidação dos jogos eletrônicos como instrumento de comunicação e de construção de conhecimento. 2004.
- [Asc07] Ascencio, Ana F. G. Método Heurístico para Projetar e Analisar Interfaces Hipermídia Inteligentes. Dezembro 2007.
- [Bag02] Bagnasco, Andrea; Chirico, Marco; Scapolla, Anna Marina. XML Technologies to Design Didactical Distributed Measurement Laboratories. *IEEE Transactions on Education*, 0-7803-7218-2/02 © 2002 IEEE, Maio 2002.
- [Bat02] Batista, Claudia R.; Ulbricht, Vânia R. Abordagem Semiótica no Desenvolvimento de Interfaces Interativas para Ambiente Hipermídia de Aprendizagem. *XIV Congreso Internacional de Ingeniería Gráfica*, Junho 2002. Espanha.
- [Boe80] Boehm, Barry W. Characteristics of Software Quality. Dezembro 1980.
- [Cas05] Castilho, Renata A. F. A Incorporação de Ambientes Virtuais de Aprendizagem no Ensino Superior. Um estudo na Universidade Estadual de Campinas. Tese de mestrado, Instituto de Artes, UNICAMP–Universidade Estadual de Campinas, 2005.
- [Che04] Chella, Marco T.;Chella; Ferreira, Elnatan C. Ferramenta para o desenvolvimento de experimentos remotos com aplicações educacionais. 2004.

- [Cob76] Cobbold, Richard S.C. Transducers for Biomedical Measurements. *IEEE Transactions on Education*, Dezembro 1976.
- [Col02] Colombo, Regina; Gerra, Ana. The Evaluation Method for Software Product. 0-7803-7218-2/02 © 2002 IEEE, Dezembro 2002.
- [Coy08] Coyne, Kara Pernice; Nielsen, Jakob. Beyond ALT Text: Making the Web Easy to Use for Users with Disabilities. Abril 2008.
- [CPq00] CPqD Telecom & IT Solutions. SAGRE. Maio 2000.
- [Dal93] Dally, James W.; Riley, William F.; McConnell, Kenneth G. Instrumentation for Engineering Measurements. Maio 1993.
- [Dem67] Deming, William Edwards; Shewhart, Walter Andrew. XML Technologies to Design Didactical Distributed Measurement Laboratories. Dezembro 1967.
- [Dow01] Downes, Stephen. Learning Objects: Resources for Distance Education Worldwide. *The International Review of Research in Open and Distance Learning*, Vol. 2, No. 1, 2001.
- [Enc07] Encarta Enciclopédia. Enciclopédia Virtual. Dezembro 2007.
- [Fil03] Filgueiras, Lucia. Engenharia de Usabilidade. Dezembro 2003.
- [HPL08] HPLABs. Memristor. *Quantum Systems*, 2008.
- [IEC07] IEC. . *IEC International Electrotechnical Commission*, Dezembro 2007.
- [ISO91] ISO/IEC. Information Technology–Software Product Evaluation–Quality characteristics and guidelines for their use, 1991. *ISO/IEC*, ISO/IEC 9126, 1991, Dezembro 1991.
- [ISO93] ISO/IEC. Ergonomic requirements for office work with visual display terminals–Part 10: dialogue design, 1993. *ISO/IEC*, ISO 9241–10, 1993, Dezembro 1993.
- [ISO97] ISO/IEC. Ergonomic requirements for office work with visual display terminals (VDT’s), Part 1: General Introduction, 1997. *ISO/IEC*, ISO/IEC 9241–1, 1997, Dezembro 1997.

- [ISO98a] ISO/IEC. Ergonomic requirements for office work with visual display terminals (VDT's). Part 11: Guidance on Usability, 1998. *ISO/IEC*, ISO 9241-11, 1998, Dezembro 1998.
- [ISO98b] ISO/IEC. Information Technology-Software Packages-Quality Requirements and Testing, 1998. *ISO/IEC*, ISO/IEC 12119, 1998, Dezembro 1998.
- [ISO98c] ISO/IEC. Information technology-Software product evaluation - Part 5: Process for evaluators, 1998. *ISO/IEC*, ISO/IEC 14598-5, 1998, Dezembro 1998.
- [ISO00] ISO/IEC. Software Engineering-Product Quality-Part 1: Quality Model, 2000. *ISO/IEC*, ISO/IEC 9126-1:2000, Dezembro 2000.
- [ISO07a] ISO. ISO International Organization for Standardization. *ISO*, Dezembro 2007.
- [ISO07b] ISO/IEC. ISO/IEC Software & System Engineering. *ISO/IEC*, Dezembro 2007.
- [Jou07] Journal. IEEE Institute of Electrical and Electronics, Inc. *IEEE*, Dezembro 2007.
- [Kan04] Kanev, F. Yu; Makenova, N.A.; Nosov, G.V. A Series of Virtual Laboratory Exercises for The Course of Theoretical Fundamentals of Electroengineering. *IEEE*, 0-7803-8383-4/04 © 2004 IEEE, Maio 2004.
- [Kil99] Kilian, Crawford. Writing for the Web. Dezembro 1999.
- [LAB07] LABIUTIL. LABIUTIL Laboratório de Utilizabilidade da Informática. Técnica de Inspeção de Conformidade Ergonômica de Software Educacional. *Laboratório de Utilizabilidade da Informática*, Dezembro 2007.
- [Lar87] Larkin, Jill H.; Simon, Herbert A. Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science*, 11:65-99, Maio 1987.
- [Lau90] Laurel, Brenda. Art of Human-Computer Interface Design. Dezembro 1990.
- [Law04] Lawenda, Marcin; Meyer, Norbert; Okon, Marcin; Rajtar, Tomasz; Stoklosa, Dominik; Kalisz, Damian; Poznan, Maciej Stroinski. Dynamic

- Measurement Scenarios in the Virtual Laboratory System. *IEEE*, Proceedings of the IEEE/ACM International Workshop on GRID Computing (GRID'04). 1550–551/04 © 2004 , COMPUTER SOCIETY, Maio 2004.
- [IU08] labvirt USP. <http://www.labvirt.futuro.usp.br/>. Technical report, USP, 2008.
- [Mar05] Mariage, Céline; Vanderdonckt, Jean; Pribeanu Costin. State of the Art of Web Usability Guidelines. Abril 2005.
- [Mar07] Marchezan, Andre R. P. Ferramentas Aplicadas no Desenvolvimento de Laboratório Remoto e/ou Presencial no Ensino de Engenharia Eletrônica. Tese de mestrado, Faculdade de Engenharia Elétrica e de Computação, UNICAMP–Universidade Estadual de Campinas, 2007.
- [MCT02] MCT. MCT–Ministério da Ciência e tecnologia. Dezembro 2002.
- [MEC07] MEC, Secretária de Educação a Distância. Utilização Pedagógica Intensiva das TIC nas Escolas. Dezembro 2007.
- [MEC08] MEC. Ministério da Educação e Cultura, 2008.
- [Men04] Mendes, Lina M. B. A vida como ela é – um projeto de utilização de outras linguagens na escola. Maio 2004.
- [Men06] Menéndez, Luis M.; Salaverría, Angel; Mandado, Enrique; Dacosta, Jacinto G. Virtual Electronics Laboratory: A New Tool to Improve Industrial Electronics Learning. *IEEE*, 1–4244–0136–4/06 © 2006 IEEE, Jun 2006.
- [Mon05] Monteiro, Silvana D. Semiótica Peirciana e a Questão da Informação e do Conhecimento. 2005.
- [Ngu01] Nguyen, Hung Q. Testing Applications on the Web: Test Planning for Internet Based System. Dezembro 2001.
- [Nie95] Nielsen, Jakob. Technology Transfer of Heuristic Evaluation and Usability Inspection. *INTERACT 95 International*, Junho 1995.
- [Nie98] Nielsen, Jakob; Loranger, Hoa. *Prioritizing Web Usability*. New Riders Press, 1998.
- [Nie02a] Nielsen, Jakob . Heuristic Evaluation. Dec 2002.

- [Nie02b] Nielsen, Jakob. Flash Usability: Design Guidelines for Web-Based Functionality, Tools, and Applications. *NORMAN GROUP*, Dezembro 2002.
- [Not95] Noth, Winfried. *Panorama da Semiótica de Platão a Peirce*. Annablume Editora, 1995.
- [Not96] Noth, Winfried. *A Semiótica no Século XX*. Annablume Editora, 1996.
- [Oli05] Oliveira, André L. L. Avaliação comparativa de diferentes modelos de interfaces gráficas empregadas no ensino de geometria, segundo os conceitos de usabilidade. Tese de mestrado, USP – Universidade Estadual de São Paulo, Novembro 2005.
- [Pag04] Pagliuso, Priscila B. B. Método Para Avaliação de Interface WEB Baseado Nos Principios da Usabilidade, AvalUWEB. Tese de mestrado, Faculdade de Engenharia Mecânica, UNICAMP, 2004.
- [Pes01] Pestana, Fábio J. Análise da Utilização do QFD – Escola de Engenharia de Itajubá, Instituto de Engenharia Mecânica – Departamento de Produção. Maio 2001.
- [Pop04] Poppel, Brept P. Van; Benson, Michael J.; Boettner, Daisie D.; Arnas A. Ozer. Virtual Laboratory Development for Undergraduate Engineering Courses. *IEEE Transactions on Education*, 0-7803-8596-9/04 © 2004 IEEE, Maio 2004.
- [Pre02] Pressman, Roger S. . *R. S. Pressman Associates, Inc*, Dezembro 2002.
- [RIV05] RIVED. Rede interativa virtual de educação, 2005.
- [Roc93] Rocha, Ana R. Um modelo para avaliação da qualidade de especificações. Tese de doutorado, PUC-Rio, Rio de Janeiro, Dezembro 1993.
- [San04] Santaella, Lucia. Games e Comunidades Virtuais. *Canal Contemporâneo*, Novembro 2004.
- [SEE08] SEED. Secretaria de Educação a Distância, 2008.
- [Sei01] Seixas, José M.; Maidantchik, Carmen . Teia de interconexão de laboratórios virtuais segundo uma concepção multidisciplinar e colaborativa. 2001.

- [Shn97] Shneiderman, Ben. Designing the User Interface Strategies for Effective Human Computer Interaction. Dezembro 1997.
- [Sil05] Silva, William R. S. Laboratórios Virtuais de Ensino a Distância com Compartilhamento de Recursos. Tese de mestrado, COPPE/UFRJ–Universidade Federal do Rio de Janeiro, M.Sc., Engenharia Elétrica, 2005.
- [Sos02] Sosteric, Nike; Hesemeier, Susan. When is a Learning Object not an Object: A first step towards a theory of learning objects. *International Review of Research in Open and Distance Learning*, Volume 3, Number 2. ISSN: 1492-3831, 2002.
- [Sou01] Souza, Antonio L.; Oliveira, José C.; Santos, Marcelo P. L. Recursos da Computação Gráfica para o Desenvolvimento de um Laboratório Virtual de Teoria Eletromagnética. *Cobenge*, Dezembro 2001.
- [Sou03] Souza, Eduardo P. Elementos Fundamentais na Melhoria da Qualidade de Software. Tese de mestrado, Faculdade de Engenharia Mecânica, UNICAMP–Universidade Estadual de Campinas, Dezembro 2003.
- [Tam02] Tambascia, Cláudia A.; Pagliuso, Priscilla. B. B.; Villas-Boas, André C. XII SRI–Seminário Regional de Informática: Lista de Verificação para validar a interação homem–computador. *XII SRI–Seminário Regional de Informática*, Novembro 2002.
- [Tog95] Tognazzini, Bruce. Tog on Software Design. Dezembro 1995.
- [UFS98] UFSC. ERGOLIST Ferramenta de Montagem do Checklist. ERGOLIST, 1998, Dezembro 1998.
- [UML07] UML® Resource Page. UML® Resource Page. Dezembro 2007.
- [Val07] Valiati, Eliane R. A.; Flores, Luciano V.; Miletto, Evandro M.; Pimenta, Marcelo S. Avaliação de Interfaces em Software Educacional: Comparando Experiências em dois Protótipos Sucessivos. Dezembro 2007.
- [Vie00] Vieira, Fabia Magali Santos. Avaliação de Software Educativo: Reflexões para uma Análise Criteriosa. Dezembro 2000.
- [Wik08] Wikipedia. Round-robin - Gerenciador de Processos. Technical report, 2008.

- [Win02] Winckler, Marco A. A.; Nemetz, Fábio; Lima, José V. Estudo de Caso da Aplicação do Método de Avaliação Heurística em um Projeto Multidisciplinar. Dezembro 2002.
- [Xao05] Xaoyan, Cui; Xiaodong, Zhang; Xi Chen . A Virtual Laboratory for Electrical and Electronics Teaching. *IEEE*, 0-7803-9128-4/05 © 2005 IEEE, April 2005.

Apêndice A

Métodos de Verificação de Usabilidade

Resumo dos Métodos de Teste de Usabilidade:

Teste de USABILIDADE é o conjunto de métodos utilizados para validação de interfaces desenvolvidas para interação com os usuários de um sistema. Busca-se descobrir qualquer problema possível de utilização do projeto. Muitos métodos de teste podem ser utilizados mesmo antes das interfaces serem completamente implementadas dentro do ciclo de desenvolvimento.

A seguir os métodos mais conhecidos:

- Heurístico (*Heuristic*): é o método mais informal que necessita de especialistas que avaliam se cada elemento de diálogo segue os princípios de USABILIDADE.
- Heurístico Estimado (*Heuristic Estimated*): é uma variação do método Heurístico no qual os avaliadores devem estimar o nível relativo de usabilidade de dois ou mais projetos em termos quantitativos (tipicamente performance esperada).
- Aprendizagem Exploratória (*Cognitive Walkthrough*): é uma técnica de avaliação do desenho de interfaces, com especial atenção para o suporte que a interface pode dar a uma aprendizagem exploratória, ou seja, a utilização pela primeira vez, sem nenhum treino prévio. O método pretende responder a uma questão: até que ponto consegue o sistema em análise guiar um utilizador não treinado na sua utilização, de modo a permitir-lhe atingir os seus objetivos?
- Avaliação por uma equipe multidisciplinar (*Pluralistic Walkthrough*): é o método em que cada grupo, composto por utilizadores, programadores e profissionais discutem cada um dos elementos da interface.
- Teste de Especificações (*Feature Inspection*): verifica a sequência de recursos necessários para se executar uma tarefa, a sequência natural, os passos desnecessários ou muito longos e se é exigido, de maneira equivocada, algum conhecimento do usuário para conclusão da tarefa.
- Teste de Consistência (*Consistency Inspection*): os projetistas verificam a aderência das implementações com seu projeto original.
- Teste Padrão (*Standards inspection*): é o método em que um especialista em interfaces padrão testa a aderência da interface.
- Teste Formal (*Formal Usability Inspection*): é o método que combina os testes individuais e em grupo em um procedimento de seis etapas com regras bem definidas que inclui elementos de testes Heurísticos e de uma forma simplificada de Aprendizagem Exploratória.

Apesar de ser a avaliação Heurística baseada na combinação dos relatórios de um conjunto de avaliadores independentes para se identificar uma lista de problemas de usabilidade, os métodos Heurístico, Heurístico Estimado, Aprendizagem Exploratória, Teste de Especificações e Teste Padrão normalmente são verificados por um único avaliador de cada vez. Em contraste, a Avaliação por uma Equipe Multidisciplinar e no Teste de Consistência os testes são efetuados por um grupo de avaliadores. Muitos métodos de teste de USABILIDADE são tão simples que permitem a utilização de simples desenvolvedores como avaliadores, muito embora os melhores resultados sejam normalmente alcançados através da avaliação de especialistas.

Apêndice B

Termos e Símbolos: Laboratório

Virtual e Usabilidade

Conceitos, Termos e Símbolos

- Action Script – Linguagem de Programação Orientada a Objetos utilizada pelo ADOBE FLASH, semelhante à linguagem JAVA.
- Ações Explícitas do Usuário – O computador deve processar somente aquelas ações solicitadas pelo usuário e apenas quando solicitado a fazê-lo. Portanto o termo “Ações Explícitas do Usuário” se refere às relações entre o processamento executado pelo computador e as ações do usuário.
- Ações Mínimas – Com o objetivo de minimizar o número de passos que o usuário deve executar, o critério “Ações Mínimas” está associado à carga de trabalho relacionada ao número de ações necessárias para a realização de uma tarefa.
- Adaptabilidade – A adaptabilidade de um sistema diz respeito a sua capacidade de reagir conforme o contexto e conforme as necessidades e preferências do usuário. Dois sub-critérios participam da adaptabilidade: a flexibilidade e a consideração da experiência do usuário.

- Agrupamento
Distinção de
Itens – O critério Agrupamento/Distinção de Itens diz respeito à organização visual dos itens de informação relacionados uns aos outros de alguma maneira. Esse critério leva em conta a localização e algumas características gráficas (formato) para indicar as relações entre os vários itens mostrados, para indicar se eles pertencem ou não a uma dada classe, ou ainda para indicar diferenças entre classes. Esse critério também diz respeito à organização dos itens de uma classe. O critério agrupamento/distinção de itens está subdividido em dois critérios: agrupamento/distinção por localização e agrupamento/distinção por formato.
- Agrupamento
Distinção por
Localização – O critério de Agrupamento/Distinção por Localização diz respeito ao posicionamento relativo dos itens, estabelecido para indicar se eles pertencem ou não a uma dada classe, ou, ainda, para indicar diferenças entre classes. Esse critério também diz respeito ao posicionamento relativo dos itens dentro de uma classe.
- Agrupamento
Distinção por
Formato – O critério de Agrupamento/Distinção por Formato diz respeito mais especificamente às características gráficas (formato, cor, etc.) que indicam se itens pertencem ou não a uma dada classe, ou indicam ainda distinções entre classes diferentes ou distinções entre itens de uma dada classe.
- Animação – Normalmente utilizada para ilustrar um acontecimento, refere-se ao conjunto de imagens que, apresentadas em seqüência, dão efeito de animação.
- Apresentação – Em um contexto interpreta-se como sendo a saída de dados de um computador para o usuário.

Apresentação Analógica	– Estilo de apresentação gráfica com grandezas e formatos que utilizam analogias com dispositivos físicos de apresentação, como velocímetros, altímetros, etc.
Apresentação Digital	– Estilo de apresentação baseada em dígitos.
Apresentação Sonora	– Apresentação de dados e informações realizadas através de dispositivos de apresentação sonora.
Área	– Região ou seção de uma tela ou janela que está localizada em uma posição consistente e é utilizada consistentemente para atingir um objetivo específico.
Artefatos Pedagógicos	– Dispositivos auxiliares utilizados no processo de aprendizagem, como o mouse, as lousas eletrônicas, projetores e outros.
Ator	– É a entidade (usuário ou sistema externo) que inicia o “caso de uso” dentro do modelo “UML”.
Atributo	– Propriedade de um objeto ou sua representação que pode ser modificado pelo usuário em determinados contextos, por exemplo, cor, tamanho, padrão ou fonte.
Barra de Rolagem	– Controle que permite ao usuário visualizar objetos que extrapolam o tamanho da área disponível para visualização.
Botão	– Figura representando botões materiais e que, normalmente, é selecionada por um dispositivo de apontamento “ <i>mouse</i> ” ou teclas de cursor, e executada por um botão do dispositivo de apontamento ou a tecla “Enter”.

- Botão de Rádio – Componente gráfico que simula os botões de um rádio real. Cada botão representa uma alternativa mutuamente exclusiva.
- Brevidade – É o critério que diz respeito à carga de trabalho perceptiva e cognitiva, tanto para entradas e saídas individuais, quanto para conjuntos de entradas isto é, conjuntos de ações necessárias para se alcançar uma meta. A Brevidade corresponde ao objetivo de limitar a carga de trabalho de leitura e entradas, e o número de passos. O critério de Brevidade está subdividido em dois critérios: Concisão e Ações Mínimas.
- Cabeçalho – Rótulo de identificação posicionado na parte superior de uma tabela ou lista.
- Caixa de agrupamento – Linhas que formam um retângulo vazado envolvendo um conjunto de objetos relacionados.
- Caixa de Diálogo – Painel que apresenta um conjunto de diferentes tipos de mostradores de dados, informações, mensagens, controles e comandos para apoiar o usuário em uma ação específica.
- Caixa de Mensagem – Nome genérico dado a qualquer caixa de diálogo que forneça informação, ou o estado corrente de um processamento em andamento, ou faça uma pergunta, ou apresente um aviso, ou chame atenção para um erro.
- Caixa de Atribuição – Os seletores são espaços na tela que são sensíveis às ações dos usuários para a seleção de valores pré-definidos pelos projetistas. Em grupos, os seletores se diferenciam entre os de escolha não exclusivas, aqui chamados de caixas de atribuição (*check-box*) e os de escolha exclusiva, ou botões de seleção (*radio button*).

-
- Campo – Campos de dados são espaços na tela que permitem ao usuário a entrada de dados e informações numéricas e alfanuméricas. Os campos são espaços sensíveis às ações de edição por parte do usuário. Os campos complexos podem ser textuais, para a entrada de linhas (mais de uma) de texto, ou gráficos, para a edição de figuras.
- Campo Op- – Campo que não precisa ser necessariamente preenchido.
cional
- Campo Obri- – Campo de um formulário que deve ser preenchido.
gatório
- Carga de Tra- – O critério Carga de Trabalho diz respeito a todos os ele-
balho mentos da interface que têm um papel importante na redução da carga cognitiva e perceptiva do usuário e no aumento da eficiência do diálogo. O critério Carga de Trabalho está subdividido em dois critérios: Brevidade (que inclui Concisão e Ações Mínimas) e Densidade Informativa.
- Caso de Uso – “*Use-Case*”, Situação ou caso real de utilização de um sistema.
- Classe – “Class”, As coisas podem ser classificadas em categorias (Automóveis, Residências, Computadores...), referidas como classes. Portanto a classe é um conjunto de coisas que possuem atributos e comportamentos similares.
- Código – Números, palavras, figuras, cores e outras formas de representar objetos e dados associados ao sistema.
- Comandos – Os comandos editáveis (linha de comando) ou selecionáveis (botão de comando) permitem a entrada de instruções do usuário que disparam funcionalidades específicas de um programa aplicativo.

- Concisão – O critério Concisão diz respeito à carga perceptiva e cognitiva de saídas e entradas individuais. Por definição, a Concisão não diz respeito às mensagens de erro e de realimentação.
- Compatibilidade – O critério compatibilidade refere-se ao grau de similaridade entre diferentes ambientes e aplicações.
- Condução – A condução refere-se aos meios disponíveis para aconselhar, orientar, informar e conduzir o usuário na interação com o computador (mensagens, alarmes, rótulos, etc.). Quatro sub-critérios participam da condução: a presteza, o agrupamento/distinção entre itens, o *feedback* imediato e a legibilidade.
- Consideração da experiência do usuário – A consideração da experiência do usuário diz respeito aos meios implementados que permitem que o sistema respeite o nível de experiência do usuário.
- Controles – Os controles são objetos sensíveis às ações dos usuários proporcionando a edição ou seleção de parâmetros que regulam os comandos.
- Controle do Usuário – O critério Controle do Usuário se refere ao fato de que os usuários deveriam estar sempre no controle do processamento do sistema (por exemplo, interromper, cancelar, suspender e continuar). Cada ação possível do usuário deve ser antecipada e opções apropriadas devem ser oferecidas.

-
- | | | |
|-----------------------|---|--|
| Controle Explícito | – | O critério Controle Explícito diz respeito tanto ao processamento explícito pelo sistema das ações do usuário, quanto ao controle que os usuários têm sobre o processamento de suas ações pelo sistema. O critério Controle Explícito se subdivide em dois critérios: Ações Explícitas do Usuário e Controle do Usuário. |
| Correção dos erros | – | O critério correção dos erros diz respeito aos meios colocados à disposição do usuário com o objetivo de permitir a correção de seus erros. |
| Critérios Ergonômicos | – | Lista de critérios e componentes para avaliação de interface homem-computador baseada nas definições feitas em 1993 por Dominique Scapin e Christian Bastien. Os critérios principais apresentados foram os seguintes: Condução, Carga de Trabalho, Controle Explícito, Adaptabilidade, Gestão de Erros, Consistência, Significado dos Códigos e Compatibilidade. |
| Cursor | – | Indicação visual do ponto corrente de entrada de dados. |
| Dado | – | Mostradores de dados são espaços na tela que apresentam ao usuário o conteúdo de registros de memória ligados às evoluções do sistema informatizado ou do sistema de referência. Eles são, por definição, insensíveis às ações do usuário. Os mostradores complexos podem organizar seus elementos de dados segundo diferentes arranjos e formas, constituindo, em particular, os mostradores de listas, de tabelas e de gráficos. |
| Default | – | Valor pré-determinado ou entrada de controle, frequentemente usado com o objetivo de reduzir as ações de entrada do usuário. |

- Densidade In- formacional – O critério Densidade Informacional diz respeito à carga de trabalho do usuário de um ponto de vista perceptivo e cognitivo, com relação ao conjunto total de itens de informação apresentados aos usuários, e não a cada elemento ou item individual.
- Diálogo – Interação entre um usuário e um sistema para atingir um determinado objetivo.
- Diagrama – É a combinação de um conjunto de elementos gráficos. O UML contém um número de elementos que combinados formam os nove tipos de diagramas. O objetivo destes diagramas é fornecer múltiplas visões de um sistema, a este conjunto de visões dá-se o nome de um modelo. O modelo UML descreve o que um sistema deve fazer mas não como fazê-lo.
- Diagrama de Classe – Descreve a representação das classes utilizada pelos desenvolvedores do sistema.
- Diagrama de Objeto – Descreve a representação de um objeto utilizada pelos desenvolvedores do sistema.
- Diagrama Caso de Uso – É a descrição de um comportamento do sistema do ponto de vista do usuário.
- Diagrama de Estado – Em um determinado momento um objeto se encontra em um estado particular. O diagrama de estado representado estes estados e suas mudanças com o tempo. Todo diagrama de estado incia com um símbolo que representa o início e outro que representa o estado final. Por exemplo, cada pessoa pode ser recém nascida, criança, adolescente ou adulta.

-
- Diagrama de Sequência – Diagrama de Classe e de Objeto são diagramas estáticos. Em um sistema em funcionamento, entretanto, os objetos interagem uns com os outros no tempo. O diagrama de sequência mostra estas interações no tempo.
- Diagrama de Atividades – Representa as atividades que ocorrem em um caso de uso relativas ao comportamento de um objeto.
- Diagrama de Colaboração – Os elementos de um sistema normalmente trabalham em conjunto afim de atingir os objetivos de um sistema. O diagrama de colaboração representa estes relacionamentos na linguagem de modelamento.
- Diagrama de Componentes – Apresenta as características de cada elemento, permitindo a sua visualização mesmo que tenha sido desenvolvido por outro membro do time de desenvolvimento.
- Diagrama de Execução – Demonstra a arquitetura física com os computadores, os dispositivos, suas conexões e o software executado em cada um.
- Dispositivo de apontamento – Dispositivo tal como “*mouse*” e “*trackball*” que permitem ao usuário mover um apontador (cursor) em um espaço de trabalho e apontar para objetos gráficos.
- Equivalente de teclado – Tecla modificadora “Control, Alt”, ou combinação de teclas (por exemplo *Control-C*) que executa uma função imediatamente, sem a necessidade de operações intermediárias. Excluem-se dessa definição as teclas de função (F1, F2, F3,...).
- Estrutura de menu – Conjunto de menus relacionados hierarquicamente (como uma estrutura de árvore) ou estruturado como uma rede.

- Execução de opção – Ação que o usuário comanda ao computador para execução de uma ou mais opções selecionadas.
- Realimentação Imediata – Diz respeito às respostas do sistema às ações do usuário. Tais entradas podem ir do simples pressionar de uma tecla até uma lista de comandos. Em todos os casos, respostas do computador devem ser fornecidas, de forma rápida, com passo “timing” apropriado e consistente para cada tipo de transação. De todo modo, uma resposta rápida deve ser fornecida com informações sobre a transação solicitada e seu resultado.
- Flexibilidade – A flexibilidade se refere aos meios colocados à disposição do usuário que lhe permitem personalizar a interface, a fim de levar em conta as exigências da tarefa, de suas estratégias ou seus hábitos de trabalho. Ela corresponde também ao número das diferentes maneiras à disposição do usuário para alcançar um certo objetivo. Trata-se, em outros termos, da capacidade da interface de se adaptar as variadas ações do usuário.
- Foco – Indicação da janela ou do objeto que receberá a entrada corrente.
- Formulário de entrada de dados – Caixa de diálogo empregado para fins de entrada de conjuntos de dados relacionados. Apresenta uma organização de diferentes tipos de dados, informações, mensagens, controles e comandos para apoiar o usuário em sua entrada de dados.
- Fonte – Conjunto de caracteres específicos, para o qual são definidas variações de tamanho e de estilo.

-
- Fonte de Tensão – Dispositivo gerador de diferença de potencial elétrico entre dois pontos de um circuito. Pode ser caracterizada ainda como de Alternada, Contínua, Quadrada, Dente de Serra e outras, segundo a variação de diferença do potencial verificada em seus terminais.
- Fonte de Corrente – Dispositivo gerador de corrente elétrica em um dado ponto de um circuito. Também pode ser caracterizada como de Alternada, Contínua, Quadrada, Dente de Serra e outras, segundo a variação de diferença do potencial verificada naquele ponto.
- Gestão de erros – A gestão de erros diz respeito a todos os mecanismos que permitem evitar ou reduzir a ocorrência de erros e, quando eles ocorrem, que favoreçam sua correção. Os erros são aqui considerados como entrada de dados incorretos, entradas com formatos inadequados, entradas de comandos com sintaxes incorretas, etc. Três sub-critérios participam da manutenção dos erros: a proteção contra os erros, a qualidade das mensagens de erro e a correção dos erros.
- GRAPPLE – “*Guidelines for Rapid APPLication Engineering*”, É o esquema de um processo de desenvolvimento que relaciona 5 etapas: Análise de Requisitos “*Requirements gathering*”, Análise “*Analysis*”, Projeto “*Design*”, Desenvolvimento “*Development*” e Testes “*Deployment*”. Cada etapa consiste de um número de ações que gera um sub-produto. Os diagramas UML se tornam sub-produtos para muitas ações.
- Grupo de opções – Grupo de opções de menu que normalmente estão relacionadas funcionalmente. Menus e painéis de menu podem conter mais de um grupo de opções.

- “Hardware” – Um dispositivo ou conjunto de partes eletrônicas estrategicamente conectados para uma aplicação desejada.
- Homogeneidade / Coerência (Consistência) – O critério homogeneidade/coerência refere-se à forma na qual as escolhas na concepção da interface (códigos, denominações, formatos, procedimentos, etc.) são conservadas idênticas, em contextos idênticos, e diferentes, em contextos diferentes.
- Ícone – Pequena imagem gráfica usada para representar uma janela ou uma função do sistema computacional.
- Identificador de opção – Código, abreviação, mnemônico, ou uma porção do nome de uma opção que é utilizada para identificar cada opção de um menu.
- Ilustração – Figura que complementa uma informação textual.
- Indicação de seleção – Qualquer característica visual que indica o elemento selecionado na tela, ao qual o usuário pode aplicar uma ação subsequente.
- Informação – Rótulos, mensagens, instruções, convites e informações sobre dados processados, que o usuário necessita utilizar para realizar as suas tarefas.
- Intermitência visual – Pisca-pisca ou recurso que define os itens que aparecem piscando nas telas.
- Item de Dado – Um conjunto de caracteres de tamanho fixo ou variável que forma uma unidade de dado simples. Por exemplo: nome, CEP (código de endereçamento postal).
- Janela – Área controlável independentemente na tela, utilizada para apresentar objetos e/ou conduzir um diálogo com o usuário.

-
- JAD – (*Joint Application Design*), No modelo UML trata-se de uma técnica aplicada desde a fase de levantamento de dados até a implantação do sistema. Normalmente associada a reuniões específicas onde se tem a representação de desenvolvedores e clientes. Seus trabalhos tem sempre como foco principal a identificação de objetivos e a modelagem conceitual. Buscam o entendimento comum das atividades e das propostas fundamentais para a realização de soluções que atendam às necessidades dos usuários do sistema.
- Legibilidade – A Legibilidade diz respeito às características lexicais das informações apresentadas na tela que possam dificultar ou facilitar a leitura dessa informação (brilho do caractere, contraste letra/fundo, tamanho da fonte, espaçamento entre palavras, espaçamento entre linhas, espaçamento de parágrafos, comprimento da linha, etc.). Por definição, o critério Legibilidade não abrange mensagens de erro ou de realimentação.
- Linguagem de comando – Tipo de diálogo no qual o usuário edita linhas de comandos que são interpretadas pelo sistema.
- Lista – Uma sucessão horizontal ou vertical de itens.
- Lista de Seleção – As listas de seleção apresentam, segundo um arranjo de lista, diversos seletores, que podem ser exclusivos ou não exclusivos.
- Manipulação Direta – Estilo de diálogo na qual o usuário age diretamente sobre o objeto na tela.
- Mapa de Menu – Representação gráfica de uma estrutura de menu.

- Menu – Conjunto de opções selecionáveis apresentadas ao usuário pelo computador. As opções pode ser apresentadas ao usuário através de dispositivos visuais (textual ou simbolicamente) ou verbais.
- Metáfora – Analogia a conceitos que já são familiares ao usuário e a partir dos quais o usuário pode derivar o uso e o comportamento do sistema.
- Modal – Atributo que define a caixa de diálogo que retém o foco das ações dos usuários. Desse modo, o usuário é obrigado a fechar a caixa modal para poder continuar seu trabalho em outras caixas ou janelas. A caixa amodal pode permanecer aberta enquanto o usuário continua trabalhando sobre outras.
- Mostrador de Dados – São espaços na tela que apresentam ao usuário o conteúdo de registros de memória ligados às evoluções do sistema informatizado ou do sistema de referência. Eles são, por definição, insensíveis às ações do usuário. Os mostradores complexos podem organizar seus elementos de dados segundo diferentes arranjos e formas, constituindo em particular os mostradores de listas, de tabelas.
- Mostrador de Gráficos – Dados especialmente formatados para mostrar relações espaciais, temporais, etc., entre conjuntos de dados.
- Mostrador de Texto – São espaços na tela que apresentam informação ou dados ocupando mais de uma linha de texto.
- Mouse – Dispositivo de apontamento normalmente utilizado em conjunto com um teclado em interfaces de usuário do tipo apontar e clicar orientadas a objeto.

-
- Navegação – Movimento de uma opção para outra dentro de um painel de menu, ou movimento de um painel de menu para outro em uma estrutura de menu.
- Níveis de experiência – Descreve as diferentes experiências de uma população de usuários. Um usuário pode, com o uso continuado, passar de aprendiz para expert no software. Ao contrário, com a falta de uso, ele pode regredir à condição de aprendiz. Esse vai-e-vem caracteriza o usuário intermitente.
- Níveis de Menu – Sub-estrutura em uma hierarquia de menu. Por exemplo, o menu inicial na hierarquia é o menu de nível 1.
- Objeto – Um objeto é um elemento de uma classe, uma coisa específica com atributos e comportamentos bem definidos.
- Objeto de Apresentação de – São objetos de software cujo processamento gera uma imagem na tela que é insensível às ações do usuário.
- Objeto de Controle de – São objetos de software que permitem a interação do usuário com o sistema informatizado. Seu processamento gera uma imagem na tela que também recebe as ações de edição textual ou gráfica ou de seleção do usuário.
- Objeto de Interação – São objetos de software que permitem a interação do usuário com o sistema informatizado. Seu processamento gera uma imagem na tela que pode também receber as ações de edição textual ou gráfica ou de seleção do usuário.
- Opção Crítica – Opção que, quando escolhida, resulta em ações que tem impacto significativo sobre o sistema ou no desempenho da tarefa, ou em ações que podem degradar significativamente o sistema ou o desempenho da tarefa.

Opção de Menu	de	–	Uma opção selecionável (textual, simbólica ou verbal) apresentada em um painel de menu.
Página de Menu	de	–	Porção de uma estrutura de menu que é apresentada ao usuário ocupando uma página inteira.
Painel de Menu	de	–	Porção de uma estrutura de menu apresentada ao usuário em uma porção específica da tela.
Presteza		–	Esse critério engloba os meios utilizados para levar o usuário a realizar determinadas ações, como, por exemplo, entrada de dados. Esse critério engloba também todos os mecanismos ou meios que permitem ao usuário conhecer as alternativas, em termos de ações, conforme o estado ou contexto nos quais ele se encontra. A presteza diz respeito igualmente às informações que permitem ao usuário identificar o estado ou contexto no qual ele se encontra, bem como as ferramentas de ajuda e seu modo de acesso.
Proteção contra os erros	con-	–	A proteção contra os erros diz respeito aos mecanismos empregados para detectar e prevenir os erros de entradas de dados, comandos, possíveis ações de conseqüências desastrosas e/ou não recuperáveis.
Qualidade das mensagens de erro	das	–	A qualidade das mensagens refere-se à pertinência, à legibilidade e à exatidão da informação dada ao usuário, sobre a natureza do erro cometido (sintaxe, formato, etc.) e sobre as ações a executar para corrigi-lo.
Rótulo		–	Etiquetas, títulos, cabeçalhos e outros elementos empregados para identificar e descrever os objetos e dados associados ao sistema. Em suas relações com outros objetos, os rótulos desempenham funções de identificação, descrição, indicação e incitação ao usuário.

-
- Seleção – Escolha de um ou mais objetos dentre um conjunto apresentado de objetos.
- Seleção de Opção – Meio através do qual o usuário indica a escolha de uma ou mais opções de um menu.
- Seleção Múltipla – Seleção de mais de uma opção em um mesmo menu antes da execução.
- Separador – Seqüência de um ou mais caracteres, linhas ou uma pausa (para voz), utilizada para separar elementos organizados.
- Significado dos códigos e denominações – O critério significado dos códigos e denominações diz respeito à adequação entre o objeto ou a informação apresentada ou pedida e sua referência. Códigos e denominações significativas possuem uma forte relação semântica com seu referente. Termos pouco expressivos para o usuário podem ocasionar problemas de condução, podendo levá-lo a selecionar uma opção errada.
- Sinal Sonoro – Bips, sirenes e sons emitidos pelo sistema para informar, chamar atenção ou dar realimentação ao usuário.
- Sistema – É uma combinação de software e hardware que provê uma solução para um problema.
- Sistema Operacional – Conjunto de programas e instruções de software que controlam os dispositivos físicos e periféricos de um computador.
- Software – Um programa ou um conjunto de programas de computadores.

Tarefa	–	Uma série de transações que compreende parte ou o todo de uma atividade do usuário.
Tecla aceleradora	–	Tecla modificadora (Control, Alt), ou combinação de teclas (por exemplo Control-C) que executa uma função imediatamente, sem a necessidade de operações intermediárias. Excluem-se dessa definição as teclas de função (F1, F2, F3,...).
Tecla de Função	–	Tecla cuja ativação afeta a entrada de controle. Por exemplo: F1, F2, F3,...
Tela	–	Todo o espaço de apresentação de um dispositivo. Uma tela inclui todas as janelas e caixas de diálogo apresentadas simultaneamente.
UML	–	“ <i>Unified Modeling Language</i> ”, é um modelo unificado que propõe a sistematização das fases de entendimento, análise, projeto, desenvolvimento e testes para a solução de um problema.
Valor Default	–	Valor pré determinado ou entrada de controle, frequentemente usado com o objetivo de reduzir as ações de entrada do usuário.
Zona Funcional	–	Área ou seção de uma tela ou janela que está localizada em uma posição consistente e é utilizada para atingir um objetivo específico
